



Microsoft Dynamics NAV 2013 SQL Readiness Training

Lab Manual for Module 3: Data Access

Microsoft®

Conditions and Terms of Use

Microsoft Confidential

This training package content is proprietary and confidential, and is intended only for users described in the training materials. This content and information is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or information included in this package is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URL and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright and Trademarks

© 2012 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Use of Microsoft Copyrighted Content** at
<http://www.microsoft.com/about/legal/permissions/>

Microsoft®, Internet Explorer®, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

About the Authors	
Author:	Gerard Conroy
Bio:	Gerard is an Escalation Engineer in the Microsoft Dynamics NAV support team based in the United Kingdom. He has been working with Microsoft Dynamics NAV since 2007 and his previous roles within Microsoft include the SQL Server support team and the internal IT Department.

Table of Contents

LAB 3: SQL TRACE FOR DATA ACCESS	1
Exercise 1: Run SQL Trace to observe Record API behavior	2
Exercise 2: Observe Connection Pooling in Action	8

Lab 3: SQL Trace for Data Access

During this lab, you will run a SQL Server Profiler trace to see how the Microsoft Dynamics NAV application creates SQL Server queries to satisfy a FIND('-') function.

Estimated time to complete this lab: **30 minutes**.

Before You Begin

The following are prerequisites before commencing with this lab:

- Complete *Module 3: Data Access*.
 - A **Microsoft Dynamics NAV 2013 Demo environment** (preferably W1)
 - Install Microsoft Dynamics NAV 2013 on your local machine.
- OR
- Create a VMAS machine using the Microsoft Dynamics NAV 2013 Demo Environment template.

The steps to do this are documented in Lab 1: Setup Considerations and Performance Tuning.

What You Will Learn

After completing this lab, you will be able to:

- Run a SQL Server Profiler trace and observe the SQL Server query which is generated when the Microsoft Dynamics NAV application executes a FIND('-') function.
- Understand how the TOP clause is used by SQL Server.

Scenario

You are developing a Microsoft Dynamics NAV 2013 application and using the FIND('-') function to retrieve results from the Microsoft Dynamics NAV database. You need to understand how SQL Server executes the SQL query, which returns the required results at the SQL Server level.

You are aware that the SQL Server Profiler tool can be used to observe the commands run by SQL Server.

Exercise 1: Run SQL Trace to observe Record API behavior

1. Start the Microsoft Dynamics NAV 2013 Developer Environment (referred to as **NAV DE** throughout the document).

If you are using the **Microsoft Dynamics NAV 2013 Public Beta** VMAS machine then the icon has been placed on the Windows Taskbar for convenience.

This will automatically open a connection to the default Microsoft Dynamics NAV database.

2. In the **NAV DE** select **Tools** and then click **Object Designer**.
3. Import the LabCode_Data Access_Lab3CU.txt object, which will be provided by your Instructor. This will create codeunit 50002.
4. After the codeunit has been imported, highlight it and press **F11** to compile it.
5. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2**, click **Performance Tools**, and then click > **SQL Server Profiler**.
6. When the SQL Server Profiler tool has started, select **File** and then select **New Trace**.
7. Connect to the SQL Server using the following parameters (See **Error! Reference source not found.**):
 - a. **Server type:** Database Engine.
 - b. **Server name:** GERARDC027 (specify your own server name here if not using VMAS).
 - c. **Authentication:** Windows Authentication.

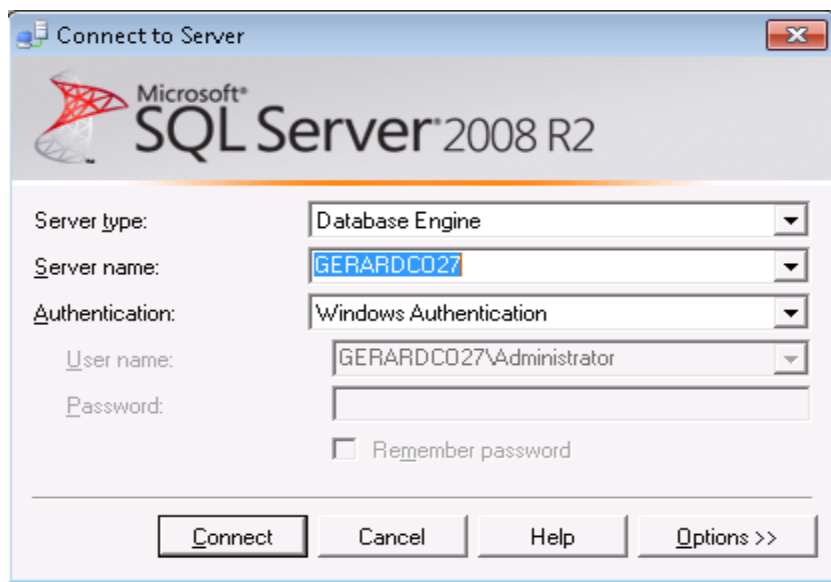


Figure 1

8. Click **Connect**.
9. Select **Event Selection** tab (see **Error! Reference source not found.**) when the **Trace Properties** window appears.

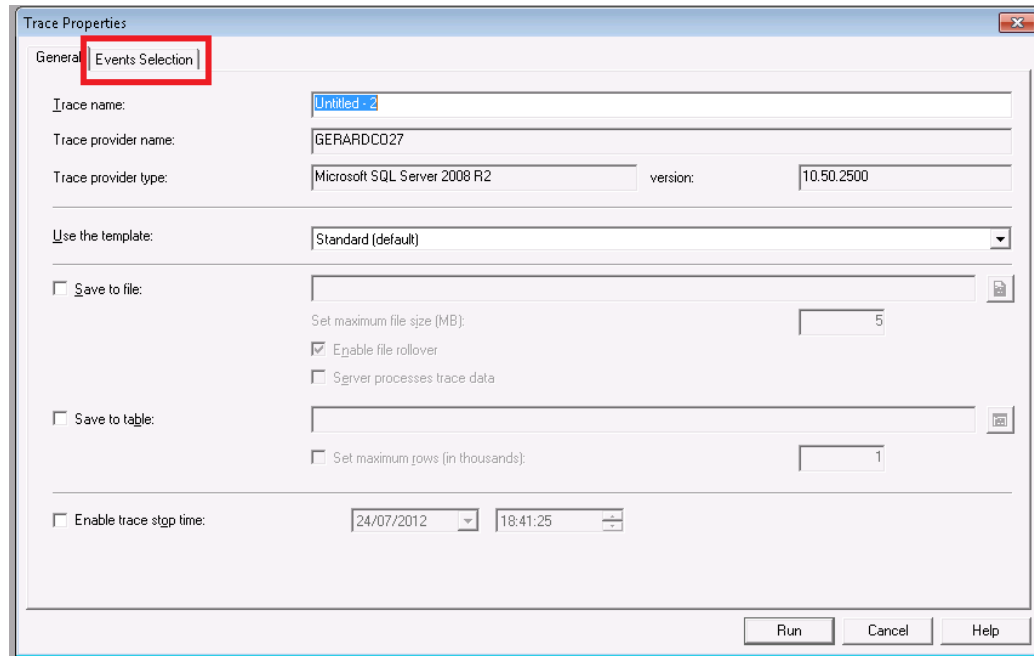


Figure 2

10. When the **Event Selection** tab is active, clear all **Events** except the **Stored Procedures – RPC: Starting** Event.

To see this event you will have to

- a. Select **Show all events** check box.
- b. Check **Show all columns** check box as indicated in **Error! Reference source not found.**

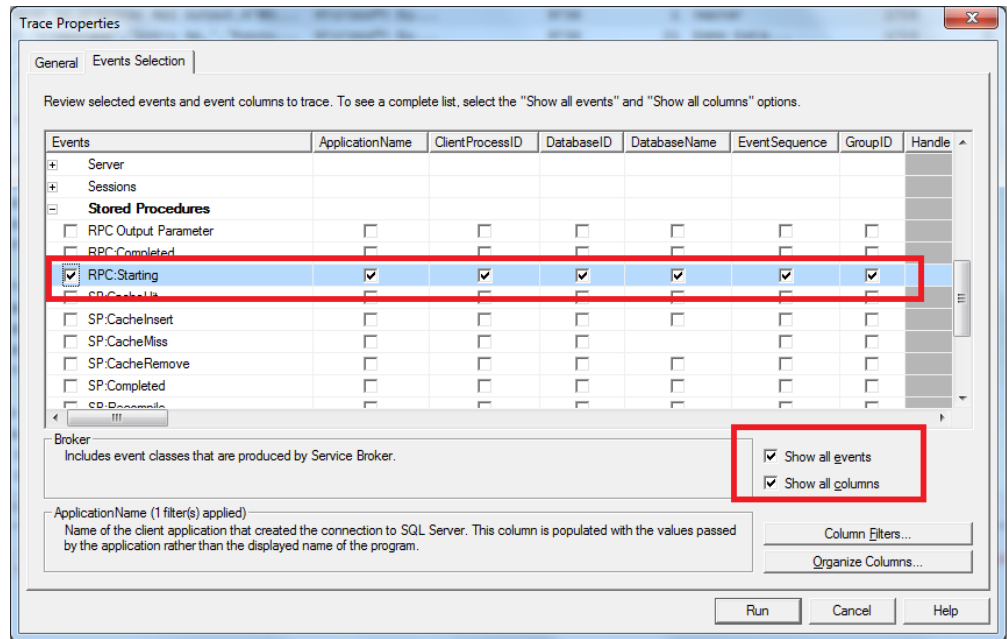


Figure 3

11. Click **Run** at the bottom of the **Trace Properties** window to start the SQL Server Profiler trace.
12. Return to the Microsoft Dynamics NAV DE environment and run codeunit 50002 (Lab3CU).

After a brief pause, the RTC will start and you will see a message like the following:

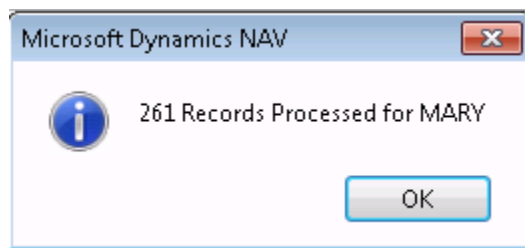


Figure 4

13. Click the red Stop button to stop the **SQL Server Profiler** trace as indicated in **Error! Reference source not found.**

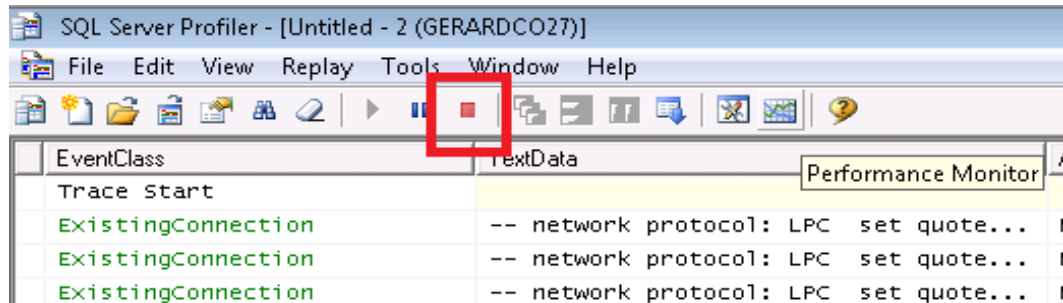


Figure 5

14. Select **Edit** and then select **Find** to search for the table we are interested in as indicated in **Error! Reference source not found.**:

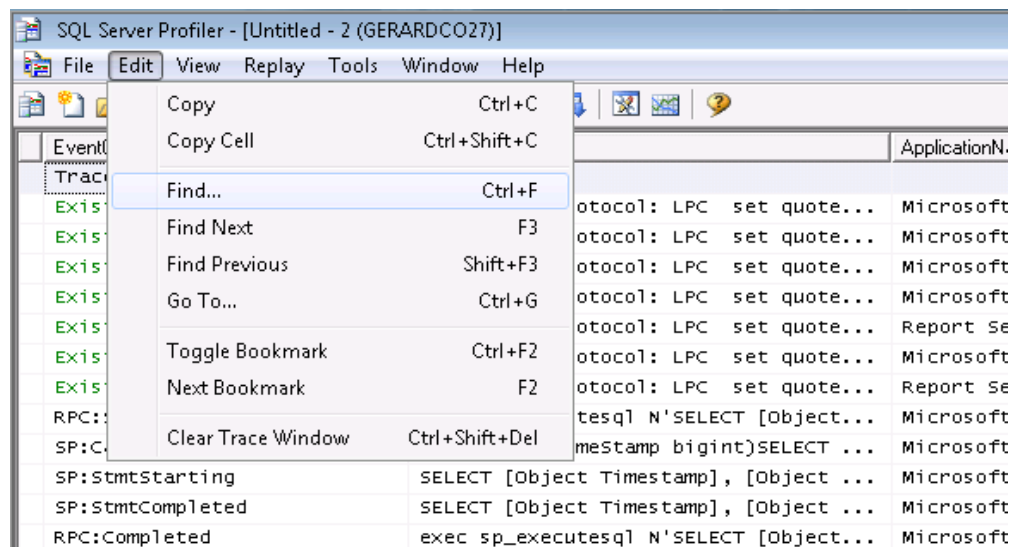


Figure 6

15. Enter **Res_Capacity Entry** under **Find What** as indicated in Figure 7:

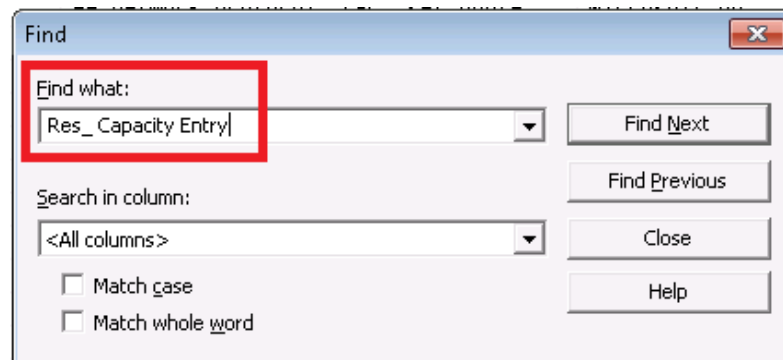


Figure 7:

16. Click **Find Next**. You should locate a record which looks like Figure 8:

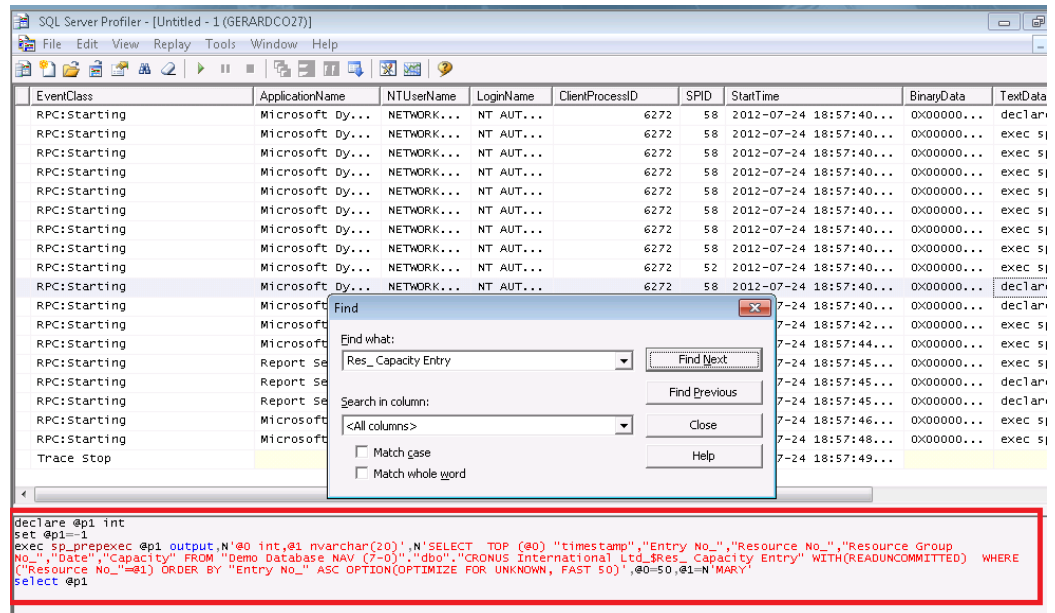


Figure 8:

17. Note the statement which is being executed:

```
declare @p1 int
set @p1=-1
exec sp_preexec @p1 output,N'@0 int,@1 nvarchar(20)',N'SELECT TOP (@0)
"timestamp", "Entry No_", "Resource No_", "Resource Group
No_", "Date", "Capacity" FROM "Demo Database NAV (7-0)". "dbo". "CRONUS
International Ltd_$Res_Capacity Entry" WITH(READUNCOMMITTED) WHERE
("Resource No_"=@1) ORDER BY "Entry No_" ASC OPTION(OPTIMIZE FOR
UNKNOWN, FAST 50)',@0=50,@1=N'MARY'
select @p1
```



Note:

Note the bits highlighted in red above. The SELECT TOP is using a variable **@0** and is no longer the constant value 50. This enables the possibility for Microsoft Dynamics NAV to tune this statement by using values less than 50 based on statistics it keeps to monitor the results from each query.

18. Select next **RPC:Starting** statement. It looks like Figure 9:

EventClass	ApplicationName	NTUserName	LoginName	ClientProcessID	SPID	StartTime	BinaryData	TextData
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	58	2012-07-24 18:57:40...	0x00000...	declare d
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	58	2012-07-24 18:57:40...	0x00000...	exec sp_e
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	58	2012-07-24 18:57:40...	0x00000...	exec sp_e
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	58	2012-07-24 18:57:40...	0x00000...	exec sp_e
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	58	2012-07-24 18:57:40...	0x00000...	exec sp_e
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	58	2012-07-24 18:57:40...	0x00000...	exec sp_e
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	52	2012-07-24 18:57:40...	0x00000...	exec sp_e
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	58	2012-07-24 18:57:40...	0x00000...	declare d
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	58	2012-07-24 18:57:40...	0x00000...	declare d
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	52	2012-07-24 18:57:42...	0x00000...	exec sp_e
RPC:Starting	Report Server	SYSTEM	NT AUT...	2308	57	2012-07-24 18:57:45...	0x00000...	exec sp_r
RPC:Starting	Report Server	SYSTEM	NT AUT...	2308	57	2012-07-24 18:57:45...	0x00000...	declare d
RPC:Starting	Report Server	SYSTEM	NT AUT...	2308	57	2012-07-24 18:57:45...	0x00000...	declare d
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	52	2012-07-24 18:57:46...	0x00000...	exec sp_e
RPC:Starting	Microsoft Dy...	NETWORK...	NT AUT...	6272	52	2012-07-24 18:57:48...	0x00000...	exec sp_e
Trace Stop						2012-07-24 18:57:49...		

```

declare @p1 int
set @p1=-1
exec sp_prepexec @p1 output,N'@0 int,@1 nvarchar(20)',N'SELECT "timestamp","Entry No_","Resource No_","Resource Group No_","Date","Capacity"
FROM "Demo Database NAV (7-0)".dbo."CRONUS International Ltd_$Res_Capacity Entry" WITH(READUNCOMMITTED) WHERE ("Entry No_">@0 AND
"Resource No_"=@1) ORDER BY "Entry No_" ASC OPTION(OPTIMIZE FOR UNKNOWN, FAST 50)',@0=50,@1=N'MARY'
select @p1

```

Figure 9:

19. This is the **SQL SELECT** statement which will return all the rows after the first 50:

```

declare @p1 int
set @p1=-1
exec sp_prepexec @p1 output,N'@0 int,@1 nvarchar(20)',N'SELECT
"timestamp","Entry No_","Resource No_","Resource Group
No_","Date","Capacity" FROM "Demo Database NAV (7-0)".dbo."CRONUS
International Ltd_$Res_Capacity Entry" WITH(READUNCOMMITTED) WHERE
("Entry No_">@0 AND "Resource No_"=@1) ORDER BY "Entry No_" ASC
OPTION(OPTIMIZE FOR UNKNOWN, FAST 50)',@0=50,@1=N'MARY'
select @p1

```

**Note:**

Note that there is no **TOP** clause in this SQL statement because all rows are retrieved if the Microsoft Dynamics NAV application reads past the 50th record in this example.

However, the **WHERE** clause ensures that only the rows subsequent to the ones returned earlier will be returned by this statement.

**More:**

For more information on how the FIND('–') statement functions in Microsoft Dynamics NAV 2013, see:

Module 3 – Data Access.

Exercise 2: Observe Connection Pooling in Action

1. Close Microsoft Dynamics NAV 2013 middle tier service.
 - a. In **Windows** go to **Start**, click **Run** and enter **services.msc**.
 - b. Locate the service named **Microsoft Dynamics NAV Server [DynamicsNAV70]**. Right-click it and select **Stop**.
2. Start **Microsoft SQL Server Management Studio** (referred to as **SSMS** throughout the document).

If you are using the *Microsoft Dynamics NAV 2013 Public Beta* VMAS machine then the icon for SSMS has been placed on the Windows Taskbar for convenience.

3. Click **Cancel** if prompted for credentials as the default connection will be for the **Object Browser**, which you do not require for this exercise.
4. Click **New Query** menu option as shown in Figure 10:

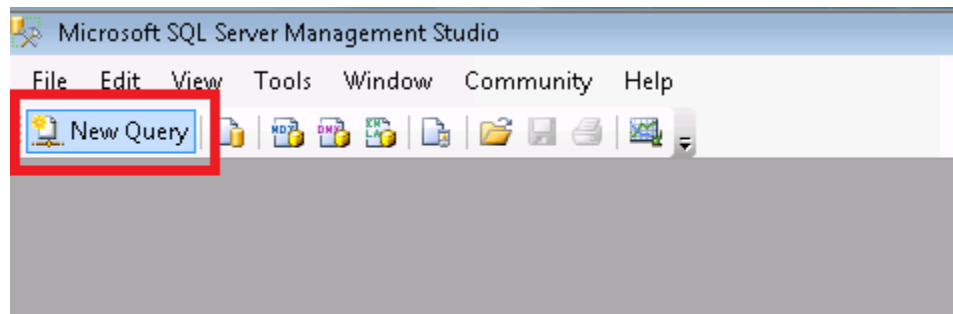


Figure 10:

5. Connect to the SQL Server instance where the Microsoft Dynamics NAV 2013 database is installed.

If you are using the **Microsoft Dynamics NAV 2013 Public Beta** VMAS machine, select the default connection settings which should be as follows:

 - a. **Server Type:** Database Engine
 - b. **Server Name:** GERARDCO27 (specify your own server name here if not using VMAS)
 - c. **Authentication:** Windows Authentication
 - d. **User Name:** GERARDCO27\Administrator (if using VMAS)
6. Click **Connect** to establish a Query session with the SQL Server, as shown in Figure 11:

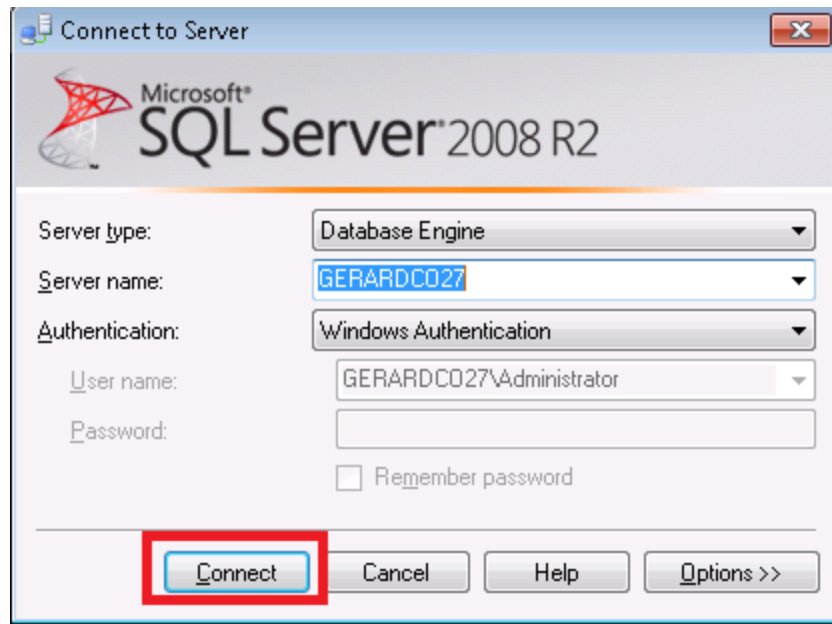


Figure 11:

7. When a Query session is established, paste the following query into the Query Window and run it by clicking **Execute** as shown in the Figure 12:

```
select spid, login_time, program_name
from master..sysprocesses
where program_name = 'Microsoft Dynamics NAV Service'
go
```

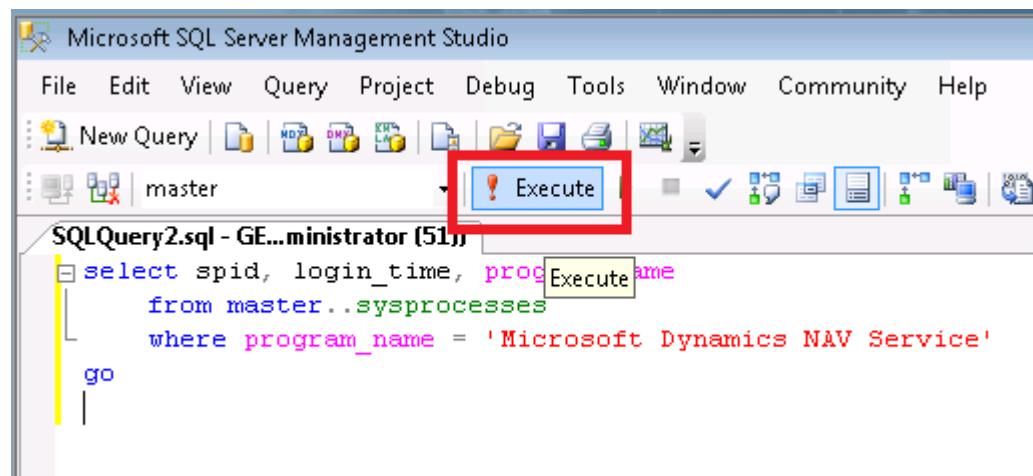


Figure 12:

8. You should get an empty result that looks similar to the following:

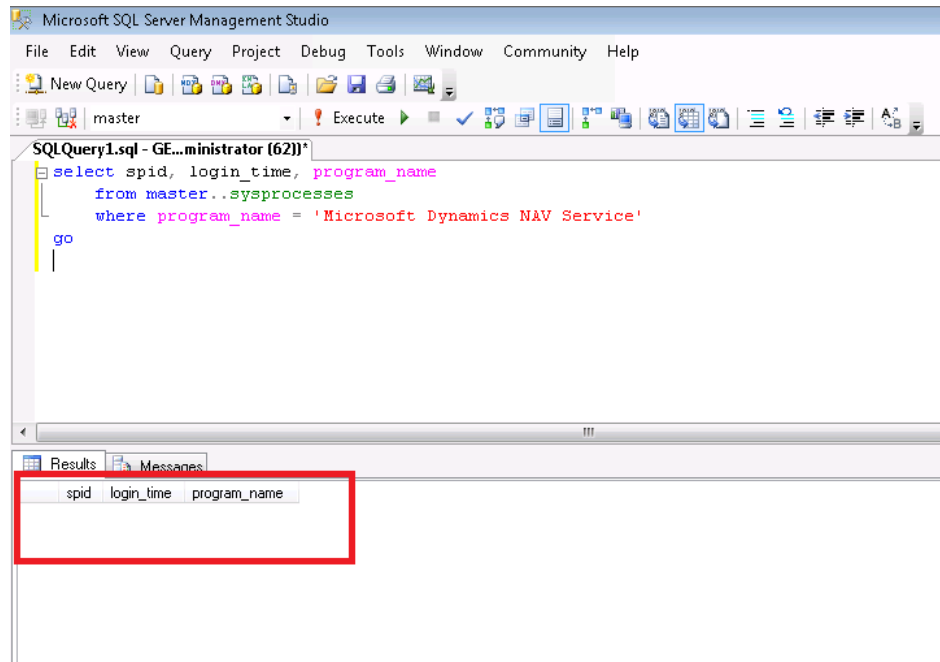


Figure 13:

You can see that there are no rows returned. This is because there are no active sessions for the Microsoft Dynamics NAV middle tier, which was shut down in step #1 as mentioned previously.

9. Start **Microsoft Dynamics NAV 2013 middle tier service**.

- a. Go to **Services** Window if it is still open. Otherwise,
 - i. Go to **Start** and then click **Run**
 - ii. Enter **services.msc**.
- b. Locate the service named **Microsoft Dynamics NAV Server [DynamicsNAV70]**. Right-click it and select **Start**.

10. Return to **SSMS** and re-run the query.

11. You will notice the results have changed to show an active middle tier connection as you would expect having started the **Microsoft Dynamics NAV Server service**.

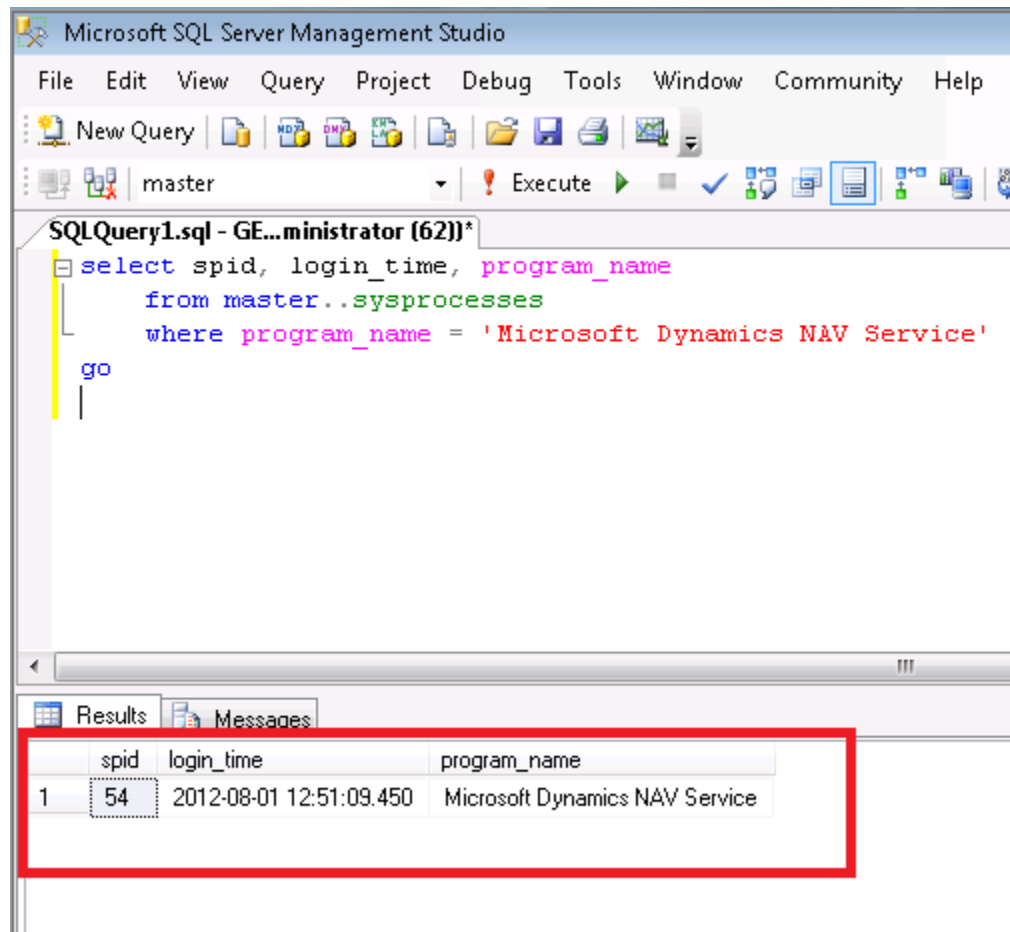


Figure 14:

This connection will not be used for Connection Pooling purposes as it is dedicated to the middle tier service

12. Start **RTC** session and when the **Role Center** is showing then run the SQL query once more.

The results will now indicate that a new middle tier connection has been established with the SQL Server. This connection will be available to the Connection pool for RTC sessions.

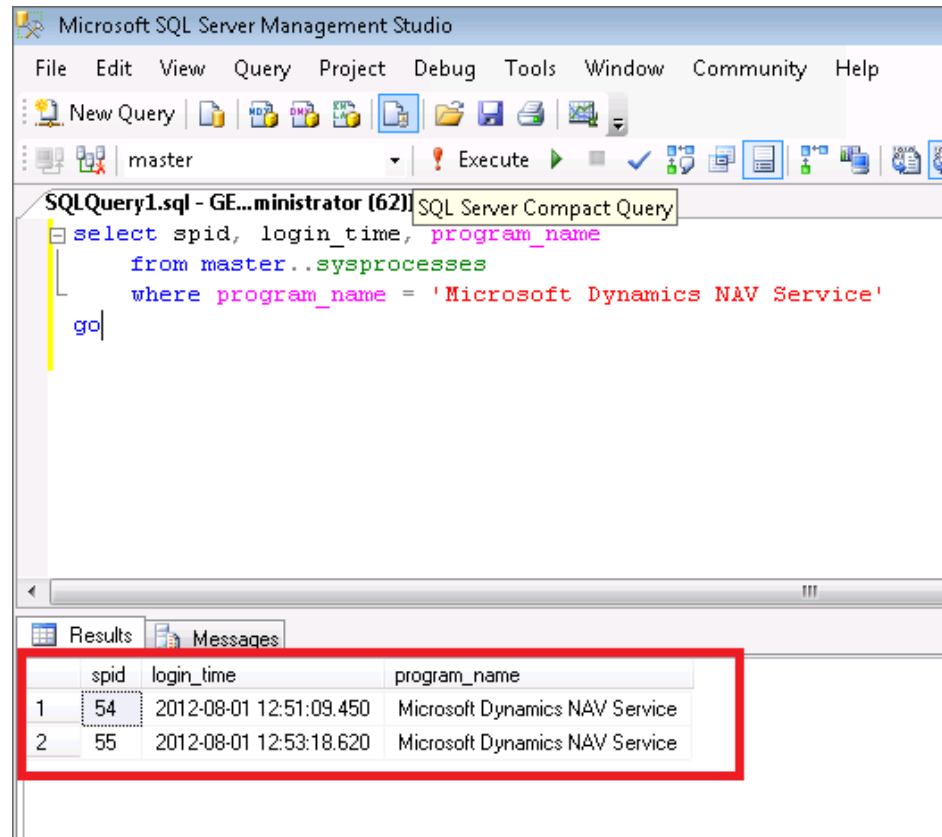


Figure 15:

13. Start a second **RTC** session and when the **Role Center** is visible run the query again.



Note:

No new middle tier connection is established in SQL Server this time, i.e., the results look the same as in Figure 6

14. Now start several new **RTC** sessions.



Note:

No new SQL Server connections are created. This is because the Microsoft Dynamics NAV application is re-using the first RTC connection for all the subsequent ones (i.e. we have a Connection Pool with just one connection in it).

If each of the RTC sessions were active simultaneously, then more connection would be added to the pool as required. However, in this case the single SQL connection is enough to satisfy the queries being generated by the multiple RTC sessions we have started.

15. If you have access to **Microsoft Dynamics NAV 2009** (or earlier), you could try the same test.

The TSQL Query you need to execute to see the connection information in SSMS would be slightly different as shown:

```
select spid, login_time, program_name
from master..sysprocesses
where program_name = 'Microsoft Dynamics NAV RTC'
```

With earlier versions of Microsoft Dynamics NAV, you would see that for every RTC session started, the middle tier service would create a new SQL Server connection.

If you have access to a Microsoft Dynamics NAV 2009 R2 environment try the previous exercise there to see the difference.