MODULE 8: MICROSOFT DYNAMICS NAV POWERSHELL

Module Overview

Microsoft Dynamics NAV 2013 includes two new tools for administering and configuring Microsoft Dynamics NAV Server and Microsoft Dynamics NAV sites:

- The Server Administration Tool
- A Windows PowerShell 2.0 API

This module describes how to use the Windows PowerShell API to configure Microsoft Dynamics NAV Server based on practical examples. It is not meant as a complete reference guide for the PowerShell commands.

Objectives

- Explain how to start the Microsoft Dynamics NAV Administration Shell.
- Explain the general commands and parameters that can be used when administering Microsoft Dynamics NAV Server.
- Illustrate how to use the PowerShell commands to administer Microsoft Dynamics NAV Server.

The PowerShell API

This lesson describes how to use the PowerShell API to administer Microsoft Dynamics NAV Server. It will explain how to start a PowerShell session and the general commands and parameters that you can use when administering Microsoft Dynamics NAV Server using Windows PowerShell API.

Lesson Objectives

- Explain how to start the Microsoft Dynamics NAV Administration Shell.
- Explain the general commands and parameters that can be used when administering Microsoft Dynamics NAV Server.

Introduction

Microsoft Dynamics NAV 2013 includes two new tools for administering and configuring Microsoft Dynamics NAV Server and Microsoft Dynamics NAV sites:

- The Server Administration Tool
- A Windows PowerShell 2.0 API

To install the tools, run Microsoft Dynamics NAV Setup and select the Server Administration Tool component.

You can use the tools to configure Microsoft Dynamics NAV Server. Please be aware that both tools do not provide the same functions for administration. Depending on the setting that you want to modify or the task that you want to perform, you select a different server administration tool.

For example, the Server Administration Tool lets you create and manage server instances. But you cannot use it to add users and permission sets or to assign permission sets to users. On the other hand, you can use the Server Administration Tool to administer server instances on remote servers, where the PowerShell API can be used only to administer server instances on the local server.

Server Administration Tool

The Server Administration Tool is a snap-in for Microsoft Management Console. When you install the Server option, the Server Administration Tool is a default feature. To run the Microsoft Dynamics NAV Server Administration Tool, click **Start > All Programs > Administrative Tools > Microsoft Dynamics NAV Server Administration**.

For more information about how to use the Server Administration Tool, see the Microsoft Dynamics NAV Server module of this course.

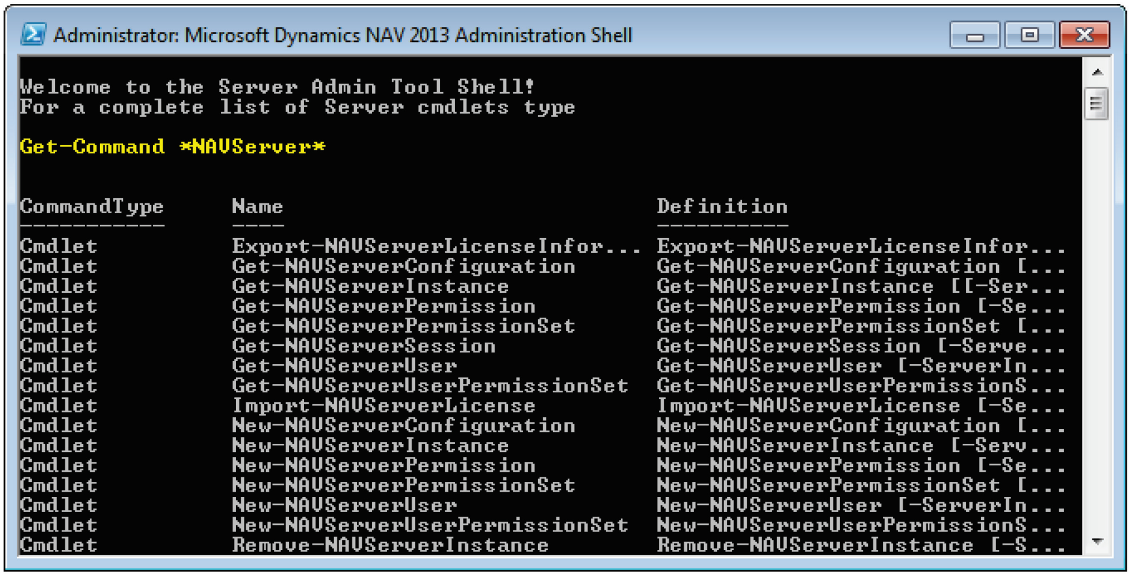
Windows PowerShell 2.0 API

A Windows PowerShell 2.0 API for administration is also available. Windows PowerShell 2.0 is included with Windows Server 2008 R2 and Windows 7. For other versions of Windows, you can install it as part of the Windows Management Framework.

Start a PowerShell Session

To start a Windows PowerShell session, click **Start > All Programs > Microsoft Dynamics NAV 2013 Administration Shell**.

When you start the session, the Administration Shell command prompt will appear:


Administrator: Microsoft Dynamics NAV 2013 Administration Shell

```
Welcome to the Server Admin Tool Shell!
For a complete list of Server cmdlets type

Get-Command *NAVServer*


CommandType     Name                             Definition
-----------     ----                             ----------
Cmdlet          Export-NAVServerLicenseInfor...  Export-NAVServerLicenseInfor...
Cmdlet          Get-NAVServerConfiguration       Get-NAVServerConfiguration [...
Cmdlet          Get-NAVServerInstance            Get-NAVServerInstance [[-Ser...
Cmdlet          Get-NAVServerPermission          Get-NAVServerPermission [-Se...
Cmdlet          Get-NAVServerPermissionSet       Get-NAVServerPermissionSet [...
Cmdlet          Get-NAVServerSession             Get-NAVServerSession [-Serve...
Cmdlet          Get-NAVServerUser                Get-NAVServerUser [-ServerIn...
Cmdlet          Get-NAVServerUserPermissionSet   Get-NAVServerUserPermissionS...
Cmdlet          Import-NAVServerLicense          Import-NAVServerLicense [-Se...
Cmdlet          New-NAVServerConfiguration       New-NAVServerConfiguration [...
Cmdlet          New-NAVServerInstance            New-NAVServerInstance [-Serv...
Cmdlet          New-NAVServerPermission          New-NAVServerPermission [-Se...
Cmdlet          New-NAVServerPermissionSet       New-NAVServerPermissionSet [...
Cmdlet          New-NAVServerUser                New-NAVServerUser [-ServerIn...
Cmdlet          New-NAVServerUserPermissionSet   New-NAVServerUserPermissionS...
Cmdlet          Remove-NAVServerInstance         Remove-NAVServerInstance [-S...
```

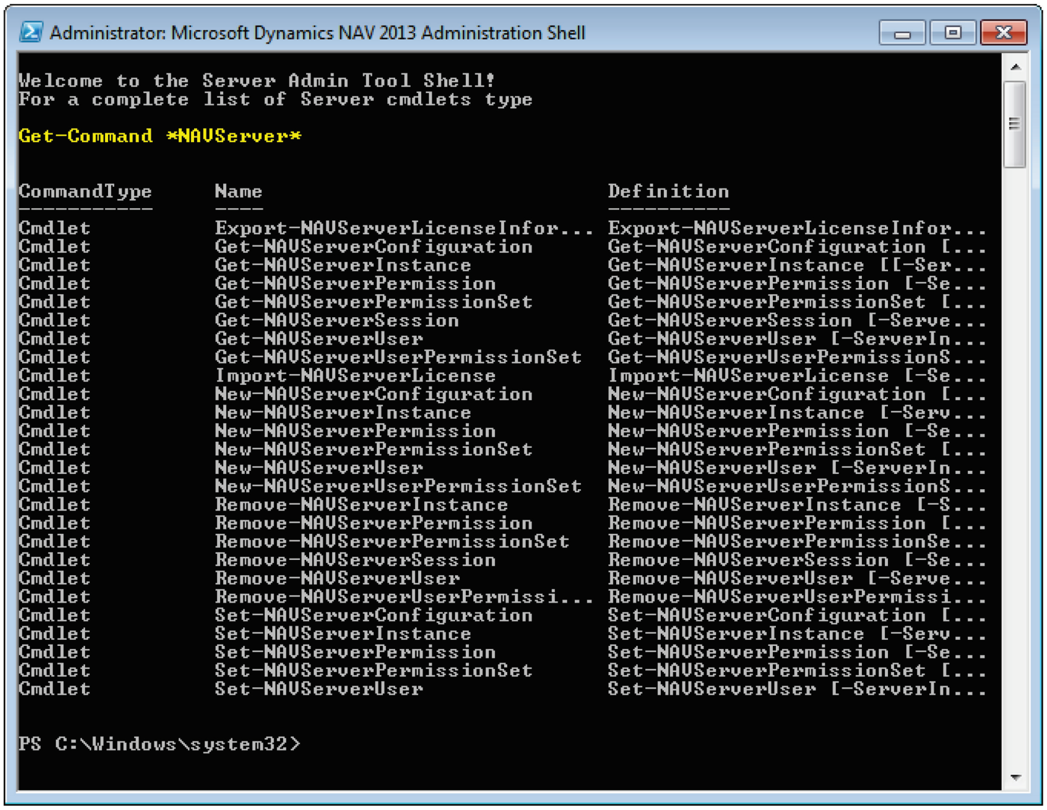
FIGURE 8.1: THE MICROSOFT DYNAMICS NAV 2013 ADMINISTRATION SHELL COMMAND PROMPT

Available Commands

To see the commands that are available for administering Microsoft Dynamics NAV, type the following command at the Windows PowerShell prompt:

```
Get-Command *NAVServer*
```

The figure, PowerShell commands for Administering Microsoft Dynamics NAV Server, shows a list of all commands that are available for administering Microsoft Dynamics NAV Server.


Administrator: Microsoft Dynamics NAV 2013 Administration Shell

```
Welcome to the Server Admin Tool Shell!
For a complete list of Server cmdlets type

Get-Command *NAVServer*


CommandType     Name                            Definition
-----------     ----                            ----------
Cmdlet          Export-NAVServerLicenseInfor... Export-NAVServerLicenseInfor...
Cmdlet          Get-NAVServerConfiguration      Get-NAVServerConfiguration [...
Cmdlet          Get-NAVServerInstance           Get-NAVServerInstance [[-Ser...
Cmdlet          Get-NAVServerPermission         Get-NAVServerPermission [-Se...
Cmdlet          Get-NAVServerPermissionSet      Get-NAVServerPermissionSet [...
Cmdlet          Get-NAVServerSession            Get-NAVServerSession [-Serve...
Cmdlet          Get-NAVServerUser               Get-NAVServerUser [-ServerIn...
Cmdlet          Get-NAVServerUserPermissionSet  Get-NAVServerUserPermissionS...
Cmdlet          Import-NAVServerLicense         Import-NAVServerLicense [-Se...
Cmdlet          New-NAVServerConfiguration      New-NAVServerConfiguration [...
Cmdlet          New-NAVServerInstance           New-NAVServerInstance [-Serv...
Cmdlet          New-NAVServerPermission         New-NAVServerPermission [-Se...
Cmdlet          New-NAVServerPermissionSet      New-NAVServerPermissionSet [...
Cmdlet          New-NAVServerUser               New-NAVServerUser [-ServerIn...
Cmdlet          New-NAVServerUserPermissionSet  New-NAVServerUserPermissionS...
Cmdlet          Remove-NAVServerInstance        Remove-NAVServerInstance [-S...
Cmdlet          Remove-NAVServerPermission      Remove-NAVServerPermission [...
Cmdlet          Remove-NAVServerPermissionSet   Remove-NAVServerPermissionSe...
Cmdlet          Remove-NAVServerSession         Remove-NAVServerSession [-Se...
Cmdlet          Remove-NAVServerUser            Remove-NAVServerUser [-Serve...
Cmdlet          Remove-NAVServerUserPermissi... Remove-NAVServerUserPermissi...
Cmdlet          Set-NAVServerConfiguration      Set-NAVServerConfiguration [...
Cmdlet          Set-NAVServerInstance           Set-NAVServerInstance [-Serv...
Cmdlet          Set-NAVServerPermission         Set-NAVServerPermission [-Se...
Cmdlet          Set-NAVServerPermissionSet      Set-NAVServerPermissionSet [...
Cmdlet          Set-NAVServerUser               Set-NAVServerUser [-ServerIn...


PS C:\Windows\system32>
```

FIGURE 8.2: POWERSHELL COMMANDS FOR ADMINISTERING MICROSOFT DYNAMICS NAV SERVER

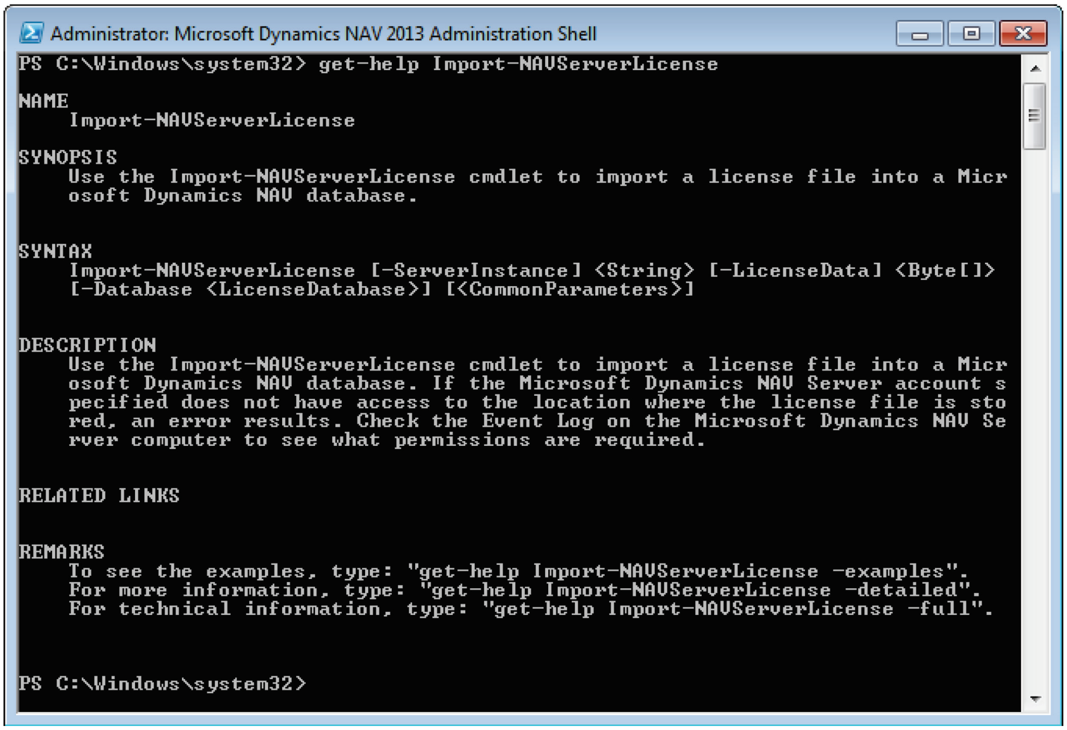
Commandlet Help

Each command has specific parameters that you can use. Some of these parameters are mandatory, some are optional. To get Help about the syntax and options for a specific command, type the following command:

```
Get-Help <command name>
```

where <*command name*> is replaced by the name of the command for which you want help.

The figure, Help Information for the Import-NAVServerLicense Command, shows available Help information for the sample command.


Administrator: Microsoft Dynamics NAV 2013 Administration Shell

```
PS C:\Windows\system32> get-help Import-NAVServerLicense

NAME
    Import-NAVServerLicense

SYNOPSIS
    Use the Import-NAVServerLicense cmdlet to import a license file into a Micr
    osoft Dynamics NAV database.


SYNTAX
    Import-NAVServerLicense [-ServerInstance] <String> [-LicenseData] <Byte[]>
    [-Database <LicenseDatabase>] [<CommonParameters>]


DESCRIPTION
    Use the Import-NAVServerLicense cmdlet to import a license file into a Micr
    osoft Dynamics NAV database. If the Microsoft Dynamics NAV Server account s
    pecified does not have access to the location where the license file is sto
    red, an error results. Check the Event Log on the Microsoft Dynamics NAV Se
    rver computer to see what permissions are required.


RELATED LINKS


REMARKS
    To see the examples, type: "get-help Import-NAVServerLicense -examples".
    For more information, type: "get-help Import-NAVServerLicense -detailed".
    For technical information, type: "get-help Import-NAVServerLicense -full".


PS C:\Windows\system32>
```

FIGURE 8.3: HELP INFORMATION FOR THE IMPORT-NAVSERVERLICENSE COMMAND

The Help information describes the meaning, the parameters and the use of the command. It describes what you can use the command for, what parameters you can use and whether the parameters are mandatory or not.

Note: *Please be aware the name of some parameters or options is case-sensitive. When you write the name of the parameters, we recommend that you use the same case as is used in the Help information. If you use the wrong case, the commands might not have the result that you want.*

Get-Help Options

By default, the Get-Help command shows basic Help information for the specified command. However, you can add some options to receive additional Help information including examples.

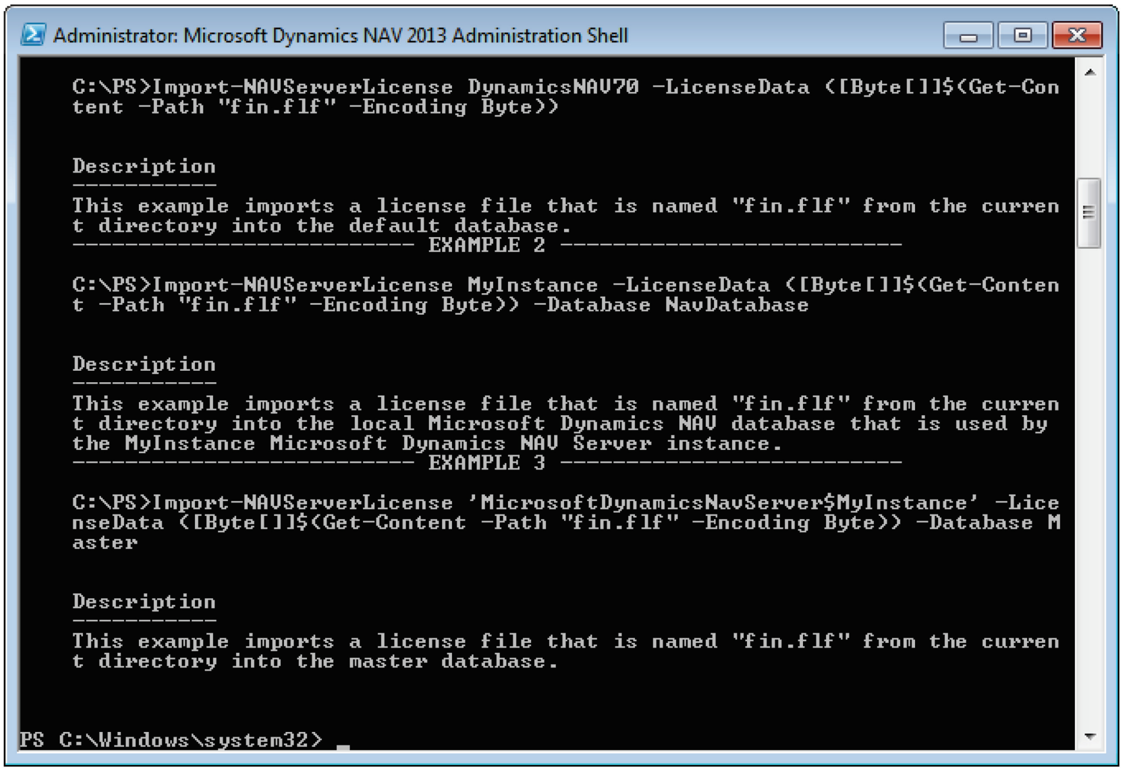
If you want to see examples for a specific command, you can use the **–examples** option.

```
Get-Help <command name> -examples
```

The following instruction shows the examples for the Import-NAVServerLicense command:

```
Get-Help Import-NAVServerLicense -examples
```

The figure, Examples for the Import-NAVServerLicense Command, shows the output of the previous command.


Administrator: Microsoft Dynamics NAV 2013 Administration Shell

```
    C:\PS>Import-NAVServerLicense DynamicsNAV70 -LicenseData ([Byte[]]$(Get-Con
    tent -Path "fin.flf" -Encoding Byte))


    Description
    -----------
    This example imports a license file that is named "fin.flf" from the curren
    t directory into the default database.
    -------------------------- EXAMPLE 2 --------------------------

    C:\PS>Import-NAVServerLicense MyInstance -LicenseData ([Byte[]]$(Get-Conten
    t -Path "fin.flf" -Encoding Byte)) -Database NavDatabase


    Description
    -----------
    This example imports a license file that is named "fin.flf" from the curren
    t directory into the local Microsoft Dynamics NAV database that is used by
    the MyInstance Microsoft Dynamics NAV Server instance.
    -------------------------- EXAMPLE 3 --------------------------

    C:\PS>Import-NAVServerLicense 'MicrosoftDynamicsNavServer$MyInstance' -Lice
    nseData ([Byte[]]$(Get-Content -Path "fin.flf" -Encoding Byte)) -Database M
    aster


    Description
    -----------
    This example imports a license file that is named "fin.flf" from the curren
    t directory into the master database.


PS C:\Windows\system32> _
```

FIGURE 8.4: EXAMPLES FOR THE IMPORT-NAVSERVERLICENSE COMMAND

To see detailed information for a command, you can add the **–detailed** option.

```
Get-Help <command name> -detailed
```

This command shows you more detailed Help information for the command. Apart from the basic information, it provides a more detailed description of each parameter (and its data type). In addition, the detailed Help information includes the examples.

Finally, you can add the **–full** option to receive technical information for the command, as shown in the following example:

```
Get-Help <command name>  -full
```

When you use this option, you receive more technical information about the command and its parameters, such as whether the parameters are mandatory or not, whether they are named or positional and whether they accept wildcard characters or not. The –full option also displays the examples.

Named parameters are parameters that always have to be specified as a "Name – Value" pair. Named parameters can typivally be entered in any order.

In the following example, the two instructions have the same result:

```
New-NAVServerInstance –ServerInstance 'ABC' –ManagementServicesPort 7045 –
ClientServicesPort 7046 –SOAPServicesPort 7047 –OdataServicesPort -7048

New-NAVServerInstance –ManagementServicesPort 7045 –ClientServicesPort
7046  –ServerInstance 'ABC' –SOAPServicesPort 7047 –OdataServicesPort 7048
```

Positional parameters are parameters that have a fixed position in the syntax. Positional parameters can be entered as a "Name – Value" pair, but you can omit the parameter name. Positional parameters must be entered in a fixed order. In many of the commands that are available in the Microsoft Dynamics NAV PowerShell session, the –ServerInstance parameter is a positional parameter. This means the following:

- Its parameter value can be specified without specifying the parameter name.
- If you do not specify the parameter name, the parameter value must always be specified immediately after the command name.

Because the ServerInstance parameter has a fixed position, the following two commands have the same result:

```
Export-NAVServerLicenseInformation –ServerInstance 'DynamicsNAV70'

Export-NAVServerLicenseInformation 'DynamicsNAV70'
```

Best Practice: *We recommend that you write the parameters as a "name - value" pair.*

Common Parameters

The common parameters are a set of parameters that you can use with any commandlet. They are implemented by Windows PowerShell, not by the commandlet developer, and they are automatically available to any commandlet.

You can use the common parameters with any commandlet, but they might not affect all commandlets. For example, if a commandlet does not generate any verbose output, the Verbose common parameter has no effect.

The –Verbose Parameter

During execution, a PowerShell command generally does not display any detailed information about the operation that is performed by the command. This information resembles the information in a trace or in a transaction log. However, if you add the –verbose parameter to a Microsoft Dynamics NAV PowerShell command, detailed information about the executed operation is displayed in the console.

For more information about this parameter (and other), enter the following instruction in a PowerShell window:

```
Get-Help about_commonparameters
```

The –AsXml Parameter

This parameter can be added to format the commandlet output as an XML node or an XML document.

Risk Mitigating Parameters

In addition to the common parameters, many commandlets offer the WhatIf and Confirm risk mitigation parameters. Commandlets that involve risk to the system or to user data usually offer these parameters.

Parameter Name	Description
-WhatIf	Displays a message that explains the operations to perform and the effect of the command, instead of executing the command.
-Confirm	Prompts you for confirmation before executing the command.

For more information about the common parameters, you can enter the following instruction at the PowerShell command prompt:

```
Get-Help about_commonparameters
```

PowerShell Commands

This lesson describes the PowerShell commands that are available to administer Microsoft Dynamics NAV Server.

Lesson Objectives

Show how to use the PowerShell commands to administer Microsoft Dynamics NAV Server.

Export-NAVServerLicenseInformation

You can use this command to export license information for the Microsoft Dynamics NAV license used by the current server instance from the database and display the license information in the console window.

The syntax for this command is as follows:

```
Export-NAVServerLicenseInformation [-ServerInstance] <string> -AsXml (Switch parameter) [<CommonParameters>]
```

The following example displays license information for the DynamicsNAV70 server instance in the console window:

```
Export-NAVServerLicenseInformation –ServerInstance ‘DynamicsNAV70’
```

Please be aware that the service account running the server instance may not have access to the database where the global license is stored. If there is an error, check the event log on the server, to see what permissions are needed.

Import-NAVServerLicense

You can use the Import-NAVServerLicense cmdlet to import a license file to the database used by the specified server instance.

The syntax for this command is as follows:

```
Import-NAVServerLicense [-ServerInstance] ‘<Server Instance Name>’ [-Database <int>] [-LicenseData <Byte[]>] [<CommonParameters>]
```

The Database parameter specifies the database to which you want to import the license file. You can use the following values:

Parameter Value	Description
1 or Master	Forces the license file to be global
2 or NavDatabase	Forces the license file to be local and stored in the NAV database used by the server instance.

The following command uploads the license file NewLicense.flf (located in the C:\Temp folder) to the DynamicsNAV70 server instance and stores the license file in the master database.

```
Import-NAVServerLicense –ServerInstance ‘DynamicsNAV70’ –LicenseData (Get-Content –Path “C:\Temp\NewLicense.flf” –Encoding Byte) –Database Master
```

The following command uploads the license file NewLicense.flf (located in the C:\Temp folder) to the DynamicsNAV70 server instance and saves the license file in the NAV database.

```
Import-NAVServerLicense –ServerInstance ‘DynamicsNAV70’ –LicenseData (Get-Content –Path “C:\Temp\NewLicense.flf” –Encoding Byte) –Database NavDatabase
```

Note: *Importing a license file to a server instance requires the server instance to be restarted in order to activate the new license. To do this, you can use the Set-NAVServerInstance command.*

Get-NAVServerSession

You can use this command to list all active (RoleTailored client for Windows and NAV Portal Framework for SharePoint) sessions that belong to a specific server instance. The commandlet uses the **Active Sessions** table to build the list.

Each session in the list has a unique id that maps to the **Session ID** field in the **Active Sessions** table.

The current PowerShell session is removed from the list.

By default, the command returns an XML document. To change the default output and display the results in the PowerShell command prompt, you can add ‘|FC’ as shown in the examples below.

The syntax for this command is as follows:

```
Get-NAVServerSession [-ServerInstance] <string> -AsXml (switch parameter)
[<CommonParameters>]
```

The following example shows the list of Windows client and SharePoint client sessions for the DynamicsNAV70 server instance and displays the list in the console.

```
Get-NAVServerSession –ServerInstance ‘DynamicsNAV70’ |fc
```

Get-NAVServerInstance

You can use the Get-NAVServerInstance cmdlet to obtain the service details of a specified server instance.

If you do not specify a server instance name, the commandlet obtains details of all registered NAV server instances on the computer that is running the server. The list contains the service name, the display name, the current state, the service account and the build number of the server instance.

The syntax for this command is as follows:

```
Get-NAVServerInstance [-ServerInstance] string –AsXml (switch parameter)
[<CommonParameters>]
```

The following example shows the service details for the DynamicsNAV70 server instance in the PowerShell console window.

```
Get-NAVServerInstance –ServerInstance ‘DynamicsNAV70’ |fc
```

The following example shows the service details for all registered Dynamics NAV server instances (on the local computer) in the PowerShell console window.

```
Get-NAVServerInstance –ServerInstance ‘’ |fc
```

Get-NAVServerPermission

You can use this command to a get list of permission sets for a specific server instance. The list can be filtered by permission set, object type or object ID. The list will show all Permission Sets that include permissions for a specific NAV object (type and number) and the effective permissions for each permission set.

The syntax for this command is as follows:

```
Get-NAVServerPermission [-ServerInstance] ‘string’ [-ObjectId <int>] [-ObjectType
<int>] [-PermissionSetId <string>] –AsXml (switch parameter)
[<CommonParameters>]
```

For the ObjectType you can use either the integer value or the text value.

Integer Value for Object Type	Text Value for Object Type
0	TableData
1	Table
2	Form
3	Report
4	Dataport
5	Codeunit
6	XMLPort
7	MenuSuite
8	Page
9	Query
10	System
11	FieldNumber

The following instruction shows a list of all permission sets that contain permissions for the Post Code table (table 225).

```
Get-NAVServerPermission –ServerInstance ‘DynamicsNAV70’ –ObjectType 0 –
ObjectId 225 |fc
```

The following instruction shows a list of all permission sets that contain permissions for the **Navigation Pane Designer** function.

```
Get-NAVServerPermission –ServerInstance ‘DynamicsNAV70’ –ObjectType System
–ObjectId 9070 |fc
```

The following instruction shows a list of all permission sets that contain permissions for codeunit 5311.

```
Get-NAVServerPermission –ServerInstance ‘DynamicsNAV70’ –ObjectType 5 –
ObjectId 5311 |fc
```

Depending on the specified parameters, the Get-NAVServerPermission command might return a large list of permission sets to the PowerShell console. In that case, the console window might not display the whole list. Instead, it uses […] (three dots) to indicate that there are more records in the list than actually shown in the console window. You can use the following command in order to display all records.

The following command displays all permission sets that contain permissions for table 18 (the Customer table) in the console window.

```
Get-NAVServerPermission –ServerInstance ‘DynamicsNAV70’ –ObjectType 0 –
ObjectId 18 |Select-xml –xpath “//Record” |fc
```

Note: *Remember to use the correct case for the parameters, and especially for the* ***Select-xml*** *option.*

Get-NAVServerPermissionSet

You can use this command to obtain a list of all Microsoft Dynamics NAV Permission Sets for a specific server instance.

The syntax for this command is as follows:

```
Get-NAVServerPermissionSet [-SeverInstance] ‘<string>’ –AsXml (switch
parameter) [<CommonParameters>]
```

The following instruction displays a list of all permission sets on the DynamicsNAV70 server instance in the console window.

```
Get-NAVServerPermissionSet [-SeverInstance] ‘DynamicsNAV70’ |fc
```

Depending on the specified parameters, the Get-NAVServerPermissionSet command might return a large list of permission sets to the PowerShell console. In that case, the console window might not display the entire list. Instead, it uses […] (three dots) to indicate that there are more records in the list than actually shown in the console window.

You can use the following command in order to display all records.

```
Get-NAVServerPermissionSet [-SeverInstance] ‘DynamicsNAV70’ |Select-xml –
xpath “//Record” |fc
```

Note: *Remember to use the correct case for the parameters, and especially for the* ***Select-xml*** *option.*

Get-NAVServerUser

You can use this command to retrieve a list of NAV users for the specified server instance.

The syntax for this command is as follows:

```
Get-NAVServerUser [-ServerInstance] ‘string’ –AsXml (switch parameter)
[<CommonParameters>] |fc
```

The following instruction lists all NAV users for the DynamicsNAV70 server instance to the console window.

```
Get-NAVServerUser –ServerInstance ‘DynamicsNAV70’ |fc
```

Get-NAVServerUserPermissionSet

This commandlet shows an overview of all permission sets per user and per company for a specific server instance. The list can be filtered by permission set, by Company Name and by Windows Account.

The syntax for the command is:

```
Get-NAVServerUserPermissionSet [-ServerInstance] ‘<Server Instance Name>’ [-
PermissionSetId ‘<string>’] [-CompanyName ‘<string>’] [-WindowsAccount
‘<string>’] –AsXml (switch parameter) [<CommonParameters>] |fc
```

The following instruction lists all permission sets that are defined on the DynamicsNAV70 server instance.

```
Get-NAVServerUserPermissionSet –ServerInstance ‘DynamicsNAV70’ |fc
```

The following instruction lists the permission sets that are assigned to the user ELLEN for the company CRONUS International Ltd.

```
Get-NAVServerUserPermissionSet –ServerInstance ‘DynamicsNAV70’ –
WindowsAccount ‘CONTOSO\ELLEN’ –CompanyName ‘CRONUS International
Ltd’.|fc
```

The following instruction lists the users who are assigned the SUPER permission set.

```
Get-NAVServerUserPermissionSet –ServerInstance ‘DynamicsNAV70’ –
PermissionSetId ‘SUPER’ |fc
```

Get-NAVServerConfiguration

You can use this command to return configuration settings for the specified Microsoft Dynamics NAV server instance. The command returns an XML document that contains the following configuration settings: NetworkType, DatabaseServer, DatabaseInstance and DatabaseName.

You can then use the Set-NAVServerConfiguration command to change the settings.

The syntax for this command is as follows:

```
Get-NAVServerConfiguration [-ServerInstance] <string> –AsXml (switch
parameter) [<CommonParameters>]
```

The following instruction returns the configuration settings for the DynamicsNAV70 server instance and displays the settings in the PowerShell session window.

```
Get-NAVServerConfiguration –ServerInstance ‘DynamicsNAV70’ |fc
```

The following instruction displays a list of all possible keys with their current values for a server instance called ‘DynamicsNAV70’.

```
 (Get-NAVServerConfiguration –ServerInstance
‘DynamicsNAV70’).Configuration.appSettings.add
```

New-NAVServerInstance

You can use the New-NAVServerInstance command to create a new NAV Server Instance, based on default values. It sets up all permissions that are needed for the server instance to start. All the configuration values for this new server instance are based on default values. Therefore one should specify DatabaseServer and DatabaseName before the server can accept client requests.

The syntax for this command is as follows:

```
New-NAVServerInstance -ClientServicesPort <int> -ManagementServicesPort
<int> -ODataServicesPort <int> -ServerInstance <string> -SOAPServicesPort
<int> [-ServiceAccount <ServiceAccount>] [-ServiceAccountCredential
<PSCredential>] [<CommonParameters>]
```

The following instruction creates a new server instance called ‘DynamicsNAV70-Test’. The server instance prompts for service account security credentials and it uses the TCP ports 7145, 7146, 7147, and 7148.

```
New-NavServerInstance –ServerInstance ‘DynamicsNAV70-Test’ –
ManagementServicesPort 7145 –ClientServicesPort 7146 –SOAPServicesPort 7147
–OdataServicesPort 7148  -ServiceAccount User
```

The following instruction creates a new server instance called ‘DynamicsNAV70-Test’. The server instance prompts for service account security credentials and it uses the TCP ports 7145, 7146, 7147, and 7148.

```
New-NavServerInstance –ServerInstance ‘DynamicsNAV70-Test’ –
ManagementServicesPort 7145 –ClientServicesPort 7146 –SOAPServicesPort 7147
–OdataServicesPort 7148  -ServiceAccount User
```

New-NAVServerPermission

You can use this commandlet to define new permissions for an existing permission set.

To create a new permission set, you first use the New-NAVServerPermissionSet command. Afterwards, you can use the New-NAVServerPermission command to add new permissions to the permission set. To remove permission from a permission set, you can use the Remove-NAVServerPermission command. To change existing object permissions for a permission set, you can use the Set-NAVServerPermission command.

The syntax for this command is as follows:

```
New-NAVServerPermission [-ServerInstance] <string>] [-Delete
<PermissionOption>] [-Execute <PermissionOption>] [-Insert
<PermissionOption>] [-Modify <PermissionOption>] [-ObjectId <int>] [-
ObjectType <int>] [-Read <PermissionOption>] [-PermissionSetId <string>] [-
SecurityFilter <string>] [<CommonParameters>]
```

PermissionSetId refers to an existing permission set.

The PermissionOption can have the following values:

Parameter Value (integer)	Parameter Value (text)	Description
0	No	Use this option if you do not want to set a specific permission type for the permission set.
1	Yes	Use this option to assign direct permissions to the permission set.

Parameter Value (integer)	Parameter Value (text)	Description
2	Indirect	Use this option to assign indirect permissions to the permission set.

You can use either the text value or the numeric value for the permission option.

For the ObjectType you can use either the integer value or the text value.

Integer Value for Object Type	Text Value for Object Type
0	TableData
1	Table
2	Form
3	Report
4	Dataport
5	Codeunit
6	XMLPort
7	MenuSuite
8	Page
9	Query
10	System
11	FieldNumber

The following instruction adds a new permission for table 5050 to the ‘ALL’ permission set on the DynamicsNAV70 server instance. The permission set contains direct read permission, direct insert permission, indirect modify permission, no delete permission and direct execute permission for table 5050.

```
New-NAVServerPermission –ServerInstance ‘DynamicsNAV70’ –PermissionSetId
‘ALL’ –ObjectType TableData –ObjectId 5050 –Read Yes –Insert Yes –Modify
Indirect –Delete No –Execute Yes
```

New-NAVServerPermissionSet

This command creates a new Microsoft Dynamics NAV permission set on the specified server instance.

The syntax for this command is as follows:

```
New-NAVServerPermissionSet [-ServerInstance] <string> -PermissionSetId
<string> -PermissionSetName <string> [<CommonParameters>]
```

PermissionSetId and PermissionSetName are mandatory parameters representing the identification code and the description for the new permission set.

Note: *We recommend that you do not use the common Microsoft Dynamics NAV filter characters (such as & and | and ..) in the name of a new permission set.*

The following instruction creates a new permission set called 'INSTCONF' on the DynamicsNAV70 server instance. The description of the permission set is 'Installation & Configuration'.

```
New-NAVServerPermissionSet –ServerInstance 'DynamicsNAV70' –PermissionSetId
'INSTCONF' –PermissionSetName 'Installation & Configuration'
```

Afterwards, you can use the New-NAVServerPermission command to add permissions to the permission set. You can use the Rename-NAVServerPermissionSet to change the name of the permission set and the Remove-NAVServerPermissionSet command to delete the permission set.

New-NAVServerUser

You can use this command to create a new Microsoft Dynamics NAV user on the specified server instance. You can either use the Windows Account or the Security Identifier (SID) to identify the user.

The Database cannot be configured to allow anonymous logons, meaning that the following accounts or SIDs cannot be used:

Security Identifier (SID)	Name
S-1-1-0	Everyone
S-1-5-7	Anonymous

The syntax for this command is as follows:

```
New-NAVServerUser [-ServerInstance] <string> [-Sid <string>] [-WindowsAccount
<string>] [<CommonParameters>]
```

You can either use the user's SID or the user's Windows Account name as a parameter. To retrieve a user's SID, you can use the Get-NAVServerUser command. Only users with credential type *WINDOWS* can be created. To create users with different credential types, you must use the RoleTailored client for Windows or the web client.

Note: *If you use the Get-NAVServerUser command to retrieve a user's SID, remember to use the value of the Windows Security ID setting, not the User Security ID.*

The following instruction adds the Windows user KEN (from the CONTOSO domain) as a user on the DynamicsNAV70 server instance.

```
New-NAVServerUser –ServerInstance 'DynamicsNAV70' –WindowsAccount
'CONTOSO\KEN'
```

The following instruction adds the Network Service account as a user to the DynamicsNAV70 instance.

```
New-NAVServerUser –ServerInstance 'DynamicsNAV70' –WindowsAccount 'NT
AUTHORITY\NETWORKSERVICE'
```

The following instruction adds the Network Service account as a user to the DynamicsNAV70 instance. Here the –Sid parameter is used instead of the WindowsAccount parameter.

```
New-NAVServerUser –ServerInstance 'DynamicsNAV70' –Sid S-1-5-20
```

New-NAVServerUserPermissionSet

You can use this command to assign a permission set to a user. If you do not specify the –CompanyName parameter, the permission set is assigned for all companies in the database. You can use the –CompanyName parameter to limit the permission set assignment to a specific company only.

The syntax for this command is as follows:

```
New-NAVServerUserPermissionSet [-ServerInstance] <string>] -PermissionSetId
<string> [-CompanyName <string>] [-Sid <string>] [-WindowsAccount <string>]
[<CommonParameters>]
```

You can either use the –Sid or the –WindowsAccount parameter to identify the user.

Please be aware that the company name is case-sensitive. Company names that contain spaces must be included in single quotation marks.

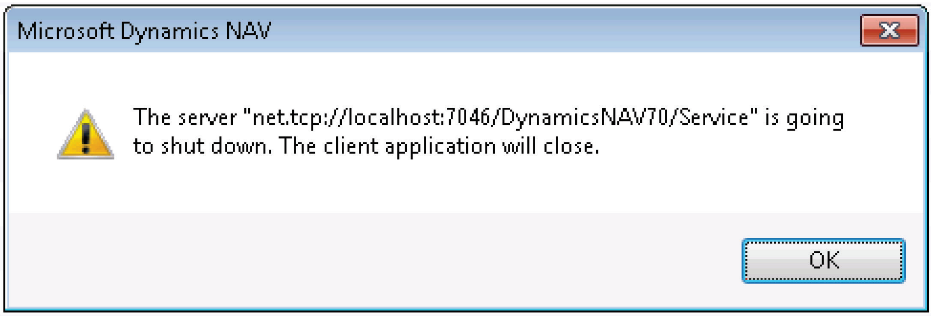
The following instruction adds the permission set named 'INSTCONF' to a user who has the Windows Account CONTOSO\KEN. The permission set is assigned for the CRONUS International Ltd. company only.

```
New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAV70' –
PermissionSetId 'INSTCONF' –CompanyName 'CRONUS International Ltd'. –
WindowsAccount 'CONTOSO\KEN'
```

Remove-NAVServerSession

You can use this command to terminate a Microsoft Dynamics NAV client session with the specified ID on the specified server instance.

You can use the Get-NAVSeverSession command to obtain a list of all active client sessions on a specific server instance. Each active client session has a Session Id. When you terminate a Windows client session, the client session is terminated and the following message is displayed on the client computer.



Microsoft Dynamics NAV

The server "net.tcp://localhost:7046/DynamicsNAV70/Service" is going to shut down. The client application will close.

FIGURE 8.5: TERMINATING A CLIENT SESSION WINDOW

The syntax for this command is as follows:

```
Remove-NAVServerSession [-ServerInstance] <string>] -SessionId <int>
[<CommonParameters>]
```

The following instruction terminates the client session with Session ID = 103 on the DynamicsNAV70 server instance.

```
Remove-NAVServerSession –ServerInstance 'DynamicsNAV70' –SessionId 103
```

Remove-NAVServerInstance

You can use this command to delete an existing Microsoft Dynamics NAV Server instance. When you use this command to remove an existing server instance, the server instance is uninstalled. This means the following:

- Folders are removed.
- Firewall exceptions are removed.
- URL access control lists are removed.
- The Windows service is unregistered.

The syntax for this command is as follows:

```
Remove-NAVServerInstance -ServerInstance <string> [<CommonParameters>]
```

The following instruction removes the DynamicsNAV70 server instance.

```
Remove-NAVServerInstance –ServerInstance ‘DynamicsNAV70’
```

Remove-NAVServerPermission

You can use this command to remove a specific permission from a permission set on a specific server instance.

The syntax for this command is:

```
Remove-NAVServerPermission [-ServerInstance] <string>] [-PermissionSetId
<string>] [-ObjectType <int>] [-ObjectId <int>] [<CommonParameters>]
```

For the ObjectType you can use either the integer value or the text value.

Integer Value for Object Type	Text Value for Object Type
0	TableData
1	Table
2	Form
3	Report
4	Dataport
5	Codeunit
6	XMLPort
7	MenuSuite
8	Page

Integer Value for Object Type	Text Value for Object Type
9	Query
10	System
11	FieldNumber

The following instruction removes permissions for Page 21 from the 'BASIC' permission set on the DynamicsNAV70 server instance.

```
Remove-NAVServerPermission –ServerInstance 'DynamicsNAV70' –
PermissionSetId 'BASIC' –ObjectType Page –ObjectId 21
```

Remove-NAVServerPermissionSet

You can use this command to remove a specific permission set from the specified server instance.

The syntax for this command is as follows:

```
Remove-NAVServerPermissionSet [-ServerInstance] <string> -PermissionSetId
<string> [<CommonParameters>]
```

The following instruction removes the INSTCONF permission set from the DynamicsNAV70 server instance.

```
Remove-NAVServerPermissionSet –ServerInstance 'DynamicsNAV70' –
PermissionSetId 'INSTCONF'
```

Remove-NAVServerUser

You can use this command to remove a Microsoft Dynamics NAV user on the specified server instance.

To identify the user to remove, you can user either the user's SID or the user's Windows Account name as a parameter. To retrieve a user's SID, you can use the Get-NAVServerUser command.

The syntax for this command is as follows:

```
Remove-NAVServerUser [-ServerInstance] <string> [-Sid <string>] [-
WindowsAccount <string>] [<CommonParameters>]
```

The following instruction removes the Network Service user from the DynamicsNAV70 server instance.

```
Remove-NAVServerUser –ServerInstance ‘DynamicsNAV70’ –WindowsAccount ‘NT
AUTHORITY\NETWORKSERVICE’
```

The following instruction removes the Network Service user from the DynamicsNAV70 server instance. It uses the –Sid instead of the –WindowsAccount parameter.

```
Remove-NAVServerUser –ServerInstance ‘DynamicsNAV70’ –Sid S-1-5-20
```

The following instruction removes the domain user KEN (from the CONTOSO domain) from the DynamicsNAV70 server instance.

```
Remove-NAVServerUser –ServerInstance ‘DynamicsNAV70’ –WindowsAccount
‘CONTOSO\KEN’
```

Remove-NAVServerUserPermissionSet

You can use this command to remove a permission set from a specific user.

The syntax for this command is as follows:

```
Remove-NAVServerUserPermissionSet [-ServerInstance] <string>] -
PermissionSetId <string> [-CompanyName <string>] [-Sid <string>] [-
WindowsAccount <string>] [<CommonParameters>]
```

To identify the user, you can user either the user’s SID or the user’s Windows Account name as a parameter. To retrieve a user’s SID, you can use the Get-NAVServerUser command.

The following instruction takes away the DOC-APP-USER permission set from the domain user CONTOSO\KEN on the DynamicsNAV70 server instance.

```
Remove-NAVServerUserPermissionSet –SeverInstance –PermissionSetID ‘DOC-
APP-USER’ –WindowsAccount ‘CONTOSO\KEN’
```

The following instruction takes away the DOC-APP-USER permission set from the domain user CONTOSO\KEN on the DynamicsNAV70 server instance. The permission set is taken away for the CRONUS International Ltd. company only (not for any other companies).

```
Remove-NAVServerUserPermissionSet –SeverInstance –PermissionSetID ‘DOC-
APP-USER’ –WindowsAccount ‘CONTOSO\KEN’ –CompanyName ‘CRONUS
International Ltd’.
```

Set-NAVServerPermissionSet

You can use this command to rename an existing permission set on the specified server instance.

The syntax for this command is as follows:

```
Set-NAVServerPermissionSet [-ServerInstance] <string> -PermissionSetId <string>
-NewPermissionSetId <string> [-PermissionSetName <string>]
[<CommonParameters>]
```

You can add the –PermissionSetName parameter if you want to change the description of the permission set. If you omit the –PermissionSetName parameter, the description of the permission set is not changed.

The following instruction renames the BASIC permission set on the DynamicsNAV70 sever instance to GENERAL. At the same time, it changes the description for the renamed permission set to ‘General Permissions’.

```
Set-NAVServerPermissionSet –ServerInstance ‘DynamicsNAV70’ –PermissionSetId
‘BASIC –NewPermissionSetId ‘GENERAL’ –PermissionSetName ‘General
Permissions’
```

Set-NAVServerUser

You can use this command to rename an existing NAV user on the specified server instance.

The syntax for this command is:

```
Set-NAVServerUser [-ServerInstance] <string> [-WindowsAccount <string>] [-Sid
<string>] [-NewWindowsAccount <string>] [<CommonParameters>]
```

To identify the users, you can use either the user’s SID or the user’s Windows account (or a combination of SID and account name).

The following instruction renames the NetworkService user to the CONTOSO\SUSAN user.

```
Set-NAVServerUser –ServerInstance ‘DynamicsNAV70’ –Sid S-1-5-20 –
NewWindowsAccount ‘CONTOSO\SUSAN’
```

The following instruction renames the domain user CONTOSO\SUSAN to the NetworkService account.

```
Set-NAVServerUser –ServerInstance ‘DynamicsNAV70’ –WindowsAccount
‘CONTOSO\SUSAN’

-NewWindowsAccount ‘NT AUTHORITY\NETWORKSERVICE’
```

Set-NAVServerInstance

You can use this command to change the service account of a specific server instance. Also it can be used to change the state of a specific server instance. In other words, you can use the command to (re)start and stop the server instance.

The syntax for this command is as follows:

```
Set-NAVServerInstance [-ServerInstance] <string> [-Domain <string>] [-Password
<string>] [-Restart] [-ServiceAccount <ServiceAccount>] [-Start] [-Stop] [-
UserName <string>] [<CommonParameters>]
```

The ServiceAccount parameter can have the following values:

Parameter Value	Description
User	Use this option if you want the server instance to run under a domain user account. If you use this option, you must specify the Domain, UserName and Password paramaters.
NetworkService	Use this option if you want the server instance to run under the ‘NT AUTHORITY\NETWORKSERVICE’ account. In this case, you do not have to specify the Domain, UserName and Password parameters.

Note: *Changing the service account for a specific server instance requires the service to be restarted before the new settings take effect. You can use the Set-NAVServerInstance with the –Restart parameter to restart the service.*

The command is typically used after you import a new license using the Import-NAVServerLicense command.

The following instruction changes service account for the DynamicsNAV70 server instance to the NetworkService account.

```
Set-NAVServerInstance –ServerInstance ‘DynamicsNAV70’ –ServiceAccount
NetworkService
```

The following instruction sets the DynamicsNAV70 server instance to run under the domain user account CONTOSO\NAV70Service account. (This example assumes that the password for this account is ‘Pa$$w0rd’.)

```
Set-NAVServerInstance –ServerInstance ‘DynamicsNAV70’ –ServiceAccount User –
Domain ‘CONTOSO’ –UserName ‘NAV70Service’ –Password ‘Pa$$w0rd’
```

The following instruction stops and restarts the DynamicsNAV70 server instance.

```
Set-NAVServerInstance –ServerInstance ‘DynamicsNAV70’ -Restart
```

Set-NAVServerPermission

You can use this command to change the permissions for a specific Permission Set on the specified server instance.

The syntax for this command is as follows:

```
Set-NAVServerPermission [[-ServerInstance] <string>] [-Delete
<PermissionOption>] [-Execute <PermissionOption>] [-Insert
<PermissionOption>] [-Modify <PermissionOption>] [-ObjectId <int>] [-
ObjectType <int>] [-Read <PermissionOption>] [-PermissionSetId <string>] [-
SecurityFilter <string>] [<CommonParameters>]
```

The PermissionOption can have the following values:

Parameter Value (integer)	Parameter Value (text)	Description
0	No	Use this option if you don’t want to set a specific permission type in the permission set.
1	Yes	Use this option to assign direct permissions to the permission set.
2	Indirect	Use this option to assign indirect permissions to the permission set.

You can use either the text value or the numeric value for the permission option.

For the ObjectType you can use either the integer value or the text value.

Integer Value for Object Type	Text Value for Object Type
0	TableData
1	Table
2	Form
3	Report
4	Dataport
5	Codeunit
6	XMLPort
7	MenuSuite
8	Page
9	Query
10	System
11	FieldNumber

The following instruction sets all permissions for the Company Information table to Indirect for the ALL permission set.

```
Set-NAVServerPermission –ServerInstance ‘DynamicsNAV70’ –PermissionSetId
‘ALL’ –Read Indirect –Insert Indirect –Modify Indirect –Delete Indirect –Execute
Indirect
```

Set-NAVServerConfiguration

You can use this command to manage the configuration settings for a specific NAV server Instance. This command is used to change configuration values for a named server instance. These values are written directly to the appSettings section in the configuration file. The changes are not applied before the server instance is restarted.

You can use the Get-NAVServerConfiguration command to obtain a list of all possible configuration keys with their values. Afterwards, you can use the Set-NAVServerConfiguration command to change the values. You can only change one configuration setting at a time.

The syntax for this command is as follows:

```
Set-NAVServerConfiguration [-ServerInstance] <string> -KeyName <string> -
KeyValue <string> [-Element <string>] [<CommonParameters>]
```

The following instruction changes the value of the “DatabaseServer” key to ‘NYC-DCSQL-2012’ for the DynamicsNAV70 server instance.

```
Set-NAVServerConfiguration –ServerInstance ‘DynamicsNAV70’ –KeyName
‘DatabaseServer’ -KeyValue  ‘NYC-DCSQL-2012’
```

Lab 8.1: Administering NAV Server Using PowerShell API

Scenario

One day Julia, the Marketing Executive who manages the sales and marketing strategy contacts Tim. Julia wants to investigate how she can use the Sales & Marketing functionality in Microsoft Dynamics NAV in order to monitor marketing effectiveness and industry success.

As a first step, Tim provides her access to the Test database, and assigns the necessary permission sets so she can fully test the Relationship Management functional area. Tim grants the permissions only for the CRONUS International Ltd. company.

Exercise 1: Check the Server Status and License

Exercise Scenario

Before he grants Julia access to the Test database, Tim checks the server state of the DynamicsNAVTest server instance. He also checks whether the correct license is used.

Task 1: Log On to the Microsoft Dynamics NAV Server Computer

High Level Steps

1. Log on to the server as Administrator.

Detailed Steps

1. Log on to the server as Administrator.
   a. On the Desktop, press Ctrl+Alt+Del.
   b. In the **Password** field, enter the password for the Administrator user:
      Passw0rd
   c. Press Enter.

Task 2: Start a PowerShell Session

High Level Steps

1. Start a Microsoft Dynamics NAV 2013 PowerShell Session.

Detailed Steps

1. Start a Microsoft Dynamics NAV 2013 PowerShell Session.
   a. In the Windows Taskbar, click **Start > All Programs > Microsoft Dynamics NAV 2013 Administration Shell**.

Task 3: Check the Server Instance State

High Level Steps

1. Retrieve a list of all registered server instances.
2. Check the status of the DynamicsNAVTest server instance.

Detailed Steps

1. Retrieve a list of all registered server instances.
   a. In the PowerShell window, enter the following: Get-NAVServerInstance |fc.

2. Check the status of the DynamicsNAVTest server instance.
   a. Check the value of the **State** setting for the DynamicsNAVTest server instance. It should read **Running**.

Task 4: Check the License Information

High Level Steps

1. Check the license information for the DynamicsNAVTest server instance.

Detailed Steps

1. Check the license information for the DynamicsNAVTest server instance.
   a. In the PowerShell window, enter the following: Export-NAVServerLicenseInformation –ServerInstance ‘DynamicsNAVTest’.
   b. Check the company information in Licensed to.

Exercise 2: Add a New User

Task 1: Add a New User to the Server Instance

High Level Steps

1. Add a new user on the DynamicsNAVTest server instance.

Detailed Steps

1. Add a new user on the DynamicsNAVTest server instance.
   a. In the PowerShell window, enter the following: New-NAVServerUser –ServerInstance ‘DynamicsNAVTest’ –WindowsAccount ‘CONTOSO\JULIA’.

Exercise 3: Assign permission sets to the User

Task 1: Add BASIC permission set

High Level Steps

1. Add BASIC permission set.

Detailed Steps

1. Add BASIC permission set.
   a. In the PowerShell window, enter the following: New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAVTest' –PermissionSetId 'BASIC' –WindowsAccount 'CONTOSO\JULIA' –CompanyName 'CRONUS International Ltd'.

Task 2: Add RM-SETUP permission set

High Level Steps

1. Add RM-SETUP permission set.

Detailed Steps

1. Add RM-SETUP permission set.
   a. In the PowerShell window, enter the following: New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAVTest' –PermissionSetId 'RM-SETUP' –WindowsAccount 'CONTOSO\JULIA' –CompanyName 'CRONUS International Ltd'.

Task 3: Add RM-CAMPAIGN, EDIT permission set

High Level Steps

1. Add RM-CAMPAIGN, EDIT permission set.

Detailed Steps

1. Add RM-CAMPAIGN, EDIT permission set.
   a. In the PowerShell window, enter the following: New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAVTest' –PermissionSetId 'RM-CAMPAIGN, EDIT' –WindowsAccount 'CONTOSO\JULIA' –CompanyName 'CRONUS International Ltd'.

Task 4: Add RM-CONT, EDIT permission set

High Level Steps

1. Add RM-CONT, EDIT permission set.

Detailed Steps

1. Add RM-CONT, EDIT permission set.
   a. In the PowerShell window, enter the following: New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAVTest' –PermissionSetId 'RM-CONT, EDIT' –WindowsAccount 'CONTOSO\JULIA' –CompanyName 'CRONUS International Ltd'.

Task 5: Add RM-OPP, EDIT permission set

High Level Steps

1. Add RM-OPP, EDIT permission set.

Detailed Steps

1. Add RM-OPP, EDIT permission set.
   a. In the PowerShell window, enter the following: New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAVTest' –PermissionSetId 'RM-OPP, EDIT' –WindowsAccount 'CONTOSO\JULIA' –CompanyName 'CRONUS International Ltd'.

Task 6: Add OSYNCHADMIN permission set

High Level Steps

1. Add OSYNCHADMIN permission set.

Detailed Steps

1. Add OSYNCHADMIN permission set.
   a. In the PowerShell window, enter the following: New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAVTest' –PermissionSetId 'OSYNCHADMIN' –WindowsAccount 'CONTOSO\JULIA' –CompanyName 'CRONUS International Ltd'.

Task 7: Add RM-PERIODIC permission set

High Level Steps

1. Add RM-PERIODIC permission set.

Detailed Steps

1. Add RM-PERIODIC permission set.
    a. In the PowerShell window, enter the following: New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAVTest' –PermissionSetId 'RM-PERIODIC' –WindowsAccount 'CONTOSO\JULIA' –CompanyName 'CRONUS International Ltd'.

Task 8: Add RM-TODO, EDIT permission set

High Level Steps

1. Add the RM-TODO, EDIT permission set.

Detailed Steps

1. Add the RM-TODO, EDIT permission set.
    a. In the PowerShell window, enter the following: New-NAVServerUserPermissionSet –ServerInstance 'DynamicsNAVTest' –PermissionSetId 'RM-TODO, EDIT' –WindowsAccount 'CONTOSO\JULIA' –CompanyName 'CRONUS International Ltd'.

Module Review

This module describes how to use the PowerShell API to administer Microsoft Dynamics NAV 2013 server instances. It explains the different PowerShell commands and their parameters. Parameters can be either named or positional.

Test Your Knowledge

Test your knowledge with the following questions.

1. What parameter can you use to obtain additional logging information during execution of a PowerShell command?

   ( ) -full

   ( ) -detailed

   ( ) -eventlog

   ( ) -showinfo

   ( ) -verbose

2. What can you use the Set-NAVServerInstance command for?

   ( ) To change the Client Services Port for a server instance.

   ( ) To enable PowerShell Management for a server instance.

   ( ) To change the service account for a server instance.

   ( ) To create a new server instance.

Test Your Knowledge Solutions

Module Review and Takeaways

1. What parameter can you use to obtain additional logging information during execution of a PowerShell command?

   ( ) -full

   ( ) -detailed

   ( ) -eventlog

   ( ) -showinfo

   (√) -verbose

2. What can you use the Set-NAVServerInstance command for?

   ( ) To change the Client Services Port for a server instance.

   ( ) To enable PowerShell Management for a server instance.

   (√) To change the service account for a server instance.

   ( ) To create a new server instance.