



NetVanta Unified Communications Technical Note

Outdial.exe Operation and Automation

Overview

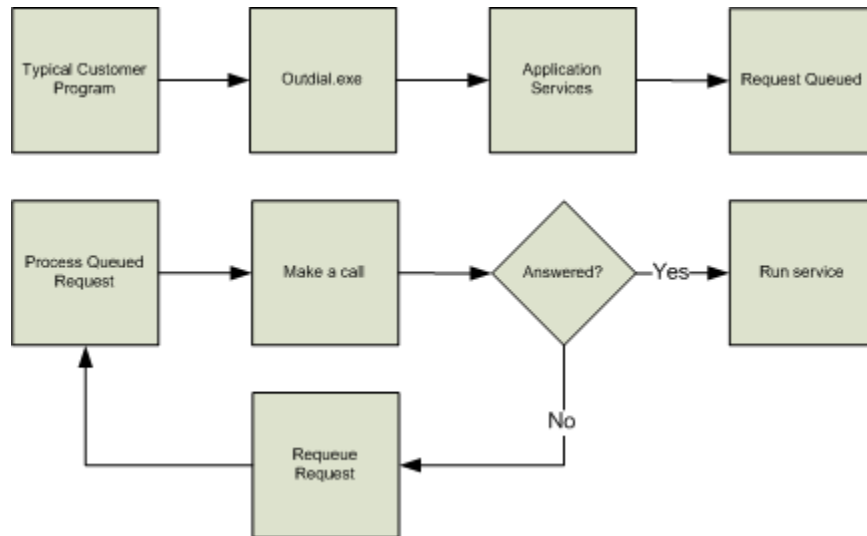
Outdial.exe is a companion application to ADTRAN's NetVanta Unified Communications product line. It is a command line program that can queue and initiate outbound calls and enable services, such as consumer notification, emergency broadcasting, appointment setting, telephone surveys, and automatic political campaign calls. Combined with database elements in a service, it is possible to record positive confirmation of a notification, and that entry in the database can be used to determine whether or not to notify the person again.

This document is intended for system integrators and IT professionals who are already familiar with the NetVanta Unified Communications product line, command line programs, Windows® Script Host, and .NET programming. It is also recommended that the IT professional have a general understanding of database principles and database design.

Basic Operation

Typically, Outdial.exe is executed using a Windows Scheduled Task, but it can also be executed manually from the command prompt. Outdial.exe recognizes an extensive list of parameters that allow customization of how calls are queued and placed.

The following figure outlines the general operation of the outbound notification feature:



Parameters

Below is a list of all of the parameters that Outdial.exe recognizes.

Parameter	What It Does	Example
-u	Specifies the user name for authentication with the server. If it is not provided, single sign-on is used. Optional unless -pw is used, but highly recommended. Currently, only the administrator account has permissions to use Outdial.exe. This includes queries and cancellations.	-u Admin
-pw	Specifies the password used for authentication with the server. Optional unless -u is used, but highly recommended.	-pw P455w0rd
-h	The server to which Outdial should connect. Required.	-h localhost -h 192.168.13.37 -h CServer.Domain.com

Parameter	What It Does	Example
-s	<p>Specifies the path to the service to execute once the call is connected.</p> <p>The path to the service is relative to the Services folder that you see in the Services pane, to the data directory for the user specified by -u. If no user is specified, then it uses single sign-on for the user's information.</p> <p>Required.</p>	<p>-s MyService</p> <p>Or</p> <p>-s "\\Shared\\Default Trunk Service"</p>
-ph	<p>The telephone numbers that are to be called. To call more than one number, separate the numbers with a comma.</p> <p>Required.</p>	<p>-ph (613)-599-9698, 6132511337</p> <p>-ph 1337</p> <p>-ph +1 (613) 599-9698</p>
-pm	<p>Specifies a list of parameters to be used by the service executed, when the call is connected. Refer to Creating UC Server Services for Outdialing on page 10 for more details.</p> <p>Optional.</p>	<p>-pm "Grant Raymond"</p> <p>-pm 9, "times",6>equals,42</p>
-sd	<p>Specifies how many time units to wait until processing the Outdial request. Time units are set with -sdu. If -sdu is omitted, the time units used are minutes.</p> <p>Default value: 0 (immediate processing).</p> <p>Optional.</p>	<p>-sd 5</p> <p>-sd 1</p> <p>-sd 42</p>
-sdu	<p>Specifies the units of the start delay specified in -sd.</p> <p>Valid values: Minute(s), Hour(s), or Day(s).</p> <p>Default value: Minutes.</p> <p>Optional.</p>	<p>-sdu minute</p> <p>-sdu minutes</p> <p>-sdu hour</p> <p>-sdu hours</p> <p>-sdu day</p> <p>-sdu days</p>
-p	<p>The number of seconds to wait after a call is connected before executing the service.</p> <p>Default value: User Message Delivery Option.</p> <p>Optional.</p>	<p>-p 5</p>
-r	<p>The number of times each number will be retried.</p> <p>Default value: User Message Delivery Option.</p> <p>Optional.</p>	<p>-r 3</p>

Parameter	What It Does	Example
-d	The number of minutes the server will wait to try again if a call goes unanswered. Default value: 3. Optional.	-d 3
-q	A query that returns a list of all Outdial requests currently queued on the server. This must be used in combination with -u , -pw , and -h .	-q (Refer to <i>Full Command Line Examples</i> below for a full example.)
-c	Cancel all Outdial requests that match the entire set of parameters used in conjunction with -c .	-c (Refer to <i>Full Command Line Examples</i> below for a full example.)

Full Command Line Examples

Queuing a Call

- This request is made with the administrator authentication to the server, *UCServer.MyDomain.com*, to call (613) 555-1234. It will make two attempts, and it will wait five minutes between attempts. When it connects, it will wait five seconds before playing *\Shared\Default Trunk Service.srv*.

```
outdial -u Admin -pw P455w0rd -h UCServer.MyDomain.com -ph 6135551234 -p 5 -d 5 -r 2 -s
"\Shared\Default Trunk Service.srv"
```

- Another request, this time with the same number, service, and authentication, but using the default pause, delay, and retry settings:

```
outdial -ph +1(613)555-1234 -s "\Shared\Default Trunk Service.srv" -pw P455w0rd -h localhost -u
Admin
```

Canceling a Queued Call

- This request will cancel the second request in the *Queuing a Call* example above. It will not cancel the first example because while the same number is reached they are formatted differently.

```
outdial -ph +1(613)555-1234 -s "\Shared\Default Trunk Service.srv" -pw P455w0rd -h localhost -u
Admin -c
```

Obtaining a List of All Queued Calls

- This sends a query request to the server located at **192.168.13.37**, using the administrator authentication:

```
outdial -q -u Admin -pw P455w0rd -h 192.168.13.37
```

- This also sends a query request, but to the server on the local machine:

```
outdial -q -pw P455w0rd -h localhost -u Admin
```

Delaying Outdial Requests

When you need an Outdial request to be made at a later time, you can delay it by a number of minutes, hours, or days. This is done using the **-sd** and **-sdu** options.

- This example will create an Outdial request that will call 6135551234 five minutes after the request is made, using the administrator account and the Shared Default Trunk service:

```
Outdial -u Admin -pw P455w0rd -h UCServer.MyDomain.com -ph 6135551234 "\Shared\Default Trunk Service.srv" -sdu minutes -sd 5
```

This does not mean that after exactly five minutes it will be sent. It means it will not be sent for five minutes. It will then behave normally.

- This example will do the same thing, but will do it two hours after the request is made:

```
Outdial -u Admin -pw P455w0rd -h UCServer.MyDomain.com -ph 6135551234 "\Shared\Default Trunk Service.srv" -sdu minutes -sd 2 -sdu hours
```

Querying and Canceling Outdial Requests

Querying Outdial Requests

When you need a list of all the outstanding Outdial requests in the queue, use the **-q** parameter, along with the server and authentication, with Outdial. This will type the requests to the command window in a format like this:

```
Outdial Query Request _____
Result:                Success
User:                  Admin
Phone Number 1:       (613)599-4757
Pause after dialing:  1
Delay between attempts: 3
Number of retries:    3
Service:               Service.srv
Time of last attempt:  on Wednesday, June 25 2008 at 01.03 PM
```

Canceling Outdial Requests

To cancel a request, the **-c** parameter is used. The user authentication (either single sign-on or **-u** and **-pw**), **-ph**, **-s**, and (if used) **-pm** parameters must be identical to how they were first entered. This can be determined by querying Outdial.

For example, the request made by:

```
Outdial -u Admin -pw password -h localhost -s Service -ph (613)599-9698
```

will not be cancelled by:

```
Outdial -u Admin -pw password -h localhost -s Service -ph 6135999698 -c
```

as the phone number parameter is formatted differently.

Server Options

There are several settings on the server that influence how Outdial behaves. Under **Tools > Options** on the **Message Delivery** tab, in the outgoing call control section, there are two options:

- The value in maximum number of concurrent calls is the number of Outdial requests and Active Message Deliveries (AMDs) that can happen at the same time. The default is **1**. Before giving this a larger value, make sure that the server and network are able to handle the increased demand for resources and bandwidth.
- The value in second(s) after connecting before prompting callee is the number of seconds the server will wait until it plays the service after the callee picks up. The default is **1**.

Changing these values will also affect AMD service. If that is undesirable, it is possible to set the pause after the callee picks up using the **-p** parameter.

Automating the Process

While it is possible to run Outdial.exe by manually entering each line, it can be automated using a Visual Basic Script and scheduling it to execute at a specific time/date with Windows Scheduled Tasks. Here are some of the key concepts of this process:

- The task can be executed based on your needs. For example, schools typically want to notify parents about their children's absences. The tool could be executed automatically every morning after attendance is taken. Another example would be a doctor's office scheduling reminder calls to be executed a day or two before patient appointments.
- The scripts usually rely on a database to make decisions, including, for example, whether or not to place a call and who to call.
- These scripts are intended to be run from NetVanta's product `.\Bin` folder. However, the script can be run from anywhere if the correct path is given to Outdial.exe and if the server has the proper permissions to run the script. Refer to [Configuring the Windows Scheduled Tasks on page 12](#).

Configuring the Components

In order to automate the process, the database, script, service, and Scheduled Task must be configured correctly. Below are the details concerning configuring these components.

Configuring the Database

Existing Database

If you already have a database, all you need to do is notate a few things:

- The names of the tables you want to access.
- The names of the columns that you want to read from the tables. Some examples are:
 - Phone numbers.
 - Qualifying fields, such as true/false “Notified?” type fields or date type fields.
 - Fields unique to each individual record, such as an employee number or serial number.
 - Any other fields you want to pass to the server with -pm.

New Database

If you’re building a new database, include the fields like the ones mentioned above. Also, if you’re using Microsoft Access, consider creating a query table as it can be accessed in the same manner and it can simplify your script later.

Example Database

If you’re using the example database, you need to change the phone numbers to valid numbers. You can use internal extensions or external numbers.

Configuring the Script

Visual Basic Scripting Edition is used in these examples. Any Microsoft .NET language, such as C++, C#, Jscript .NET, or Visual Basic .NET, could be used.

Example VBScript

Included in this package is a sample script file named **example.vbs**. Successfully running this specific script relies on three things:

- The Microsoft Access database named Simple.mdb located in *C:\DataSources\Simple.mdb*. (If you change strDBLocation, you may rename it or store it elsewhere.)
- A service named Simple.srv in the administrator’s service folder. (The service may be named anything, as long as the script is changed accordingly.)
- The strPassword variable must be set to the current password of the administrator account.

Below is a table of all the variables in SimpleExample.vbs, to give you an idea of what you can do with your own script. All of these variables are located at the top of the file, and are commented for editing ease. The code is also in [Appendix A](#), located at the end of this technical note. There is also a script in [Appendix A](#) that uses no database. Its usefulness is limited, but it is easy to follow.

Variable	What It Does	Example
<i>strDBLocation</i>	Contains the path to the database to be accessed.	strDBLocation = C:\Databases\simple.mdb
<i>strTableName</i>	Contains the name of the table you want to access. In the scripts where multiple tables are accessed, they are numbered or descriptively named.	strTableName1 = Table1 strBookTable = Classics
<i>strPhoneField</i>	Contains the name of the column where the phone numbers are stored. If they are stored in one field separated by commas, no further action is needed. If the numbers are in separate fields, they need to be numbered, and commas need to be added to the Dialout function, where appropriate.	strPhoneField = PhoneNumber OR strPhoneField1 = Cell strPhoneField2 = Home strPhoneField3 = Work
<i>strParameterX</i> (where X is an integer between 1 to 10)	<p>Contains the name of a column where any text or integer value is stored in the database. In the script, that data is passed to the server as ParameterX, which can be used in your services. The most common use is to store data unique to each record in the database, such as an employee number or a full name, which can be used by the server to identify this record.</p> <p>One could also just load the data the server will need into the parameters, and then configure the service to use the parameters instead of loading a record from the database. The downside to this method is that it makes it harder for the server to update the database, and only 10 parameters can be passed. All the data passed to the server by Outdial is in string format. If <i>strParameter1</i> holds a column title for a column that holds graphics or faxes, an error will occur.</p>	<p>strParameter1 = Employee ID strParameter2 = Date strParameter3 = Time strParameter4 = John West strParameter5 = Peter Reid strParameter6 = Jason Viets strParameter7 = Tomorrow strParameter8 = 123456789</p> <p>Parameters 1 to 3 contain column names for record identification.</p> <p>Parameters 4 to 8 could be spoken aloud using the text to speech (TTS) element.</p>

Variable	What It Does	Example
<i>strHostName</i>	Contains the network location of the server.	strHostName = "localhost" strHostName = "192.168.13.37" strHostName = "UCS.Domain.com"
<i>strServiceLoc</i>	Specifies the service that call recipients will hear when connected. The path starts in the Services folder associated with the user name and password used. To access Service.srv in the Services folder, enter a value of Service.srv. To use the Default Trunk Service in the Shared folder, enter a value of <i>\Shared\Default Trunk Service</i> .	strServiceLoc = Service.srv strServiceLoc = \Shared\Default Trunk Service.srv
<i>strUserName</i>	Contains the user name used to authenticate the request with the UC server.	strUserName = Admin
<i>strPassword</i>	Contains the password used to authenticate the request with the UC server.	strPassword = P455w0rd

Custom VBScripts

Generally, script files that access a database and run Outdial.exe follow a basic pattern.

1. Optional, but recommended. Declare, then define variables for:
 - Database manipulation and access
 - Server authentication
 - Parameters for Outdial.exe
2. Optional, but recommended. Create a function that will build the command line, then execute Outdial.exe with the parameters.
3. Open a connection to the database.
4. Select records (possibly depending on a specific value in each record) from a specific table.
5. For each record (possibly depending on another specific value), build and execute an Outdial.exe command line string.
6. Close the connection to the database.

Steps 4 and 5 can change depending on what the script is attempting to accomplish:

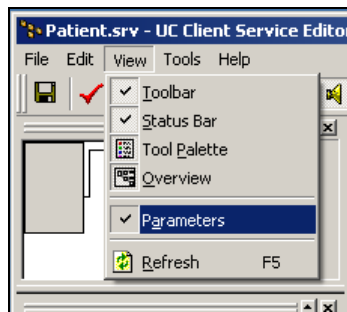
- If two tables are being accessed, Step 4 could occur for each table or a different kind of data request could be used to get data from two tables at once.
- Step 5 can be performed in a loop: *Do...While* or *For Each ... Next* statements, which could have qualifying *If* statements inside or outside the loops, and that could all be nested as deeply as necessary.

NOTE: To simplify running scripts, run this command: ***cscript //H:CScript //NoLogo //S***. It will cause *cscript* to become the default script host, suppress the Microsoft logo banner, and save the setting.

Creating UC Server Services for Outdialing

Creating services for outdialing is no different from creating other services, except for a few simple and powerful additions.

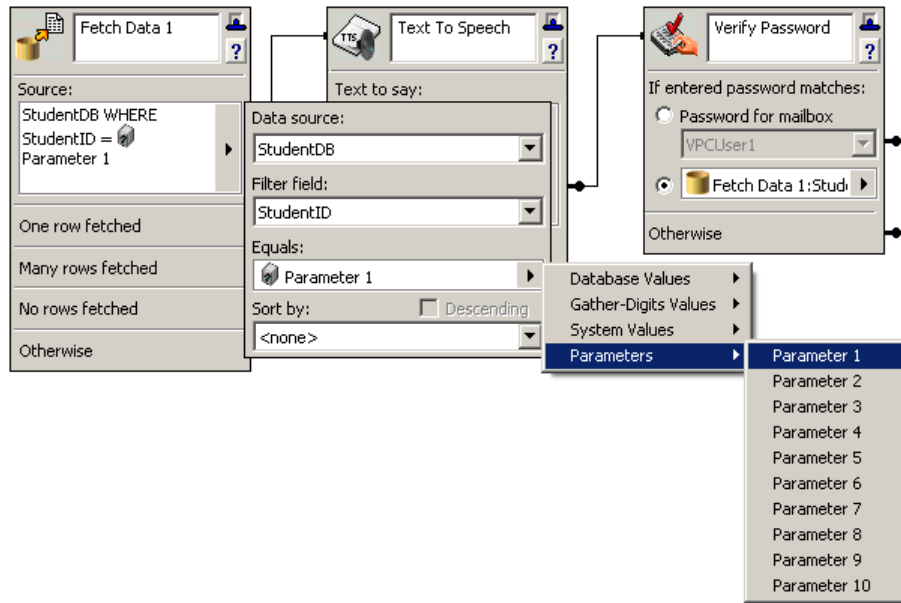
In Version 4.2 of the service creation environment, you have access to parameters. There are 10 parameters and you can use them in several different places. In order to use them in your service, you must first enable them by selecting **View > Parameters**.



Here are several guidelines for using parameters:

- Parameters may be used in the **Equals** field in the **Data Source Selector** element (any element that deals with a database).
- The **Assisted Transfer**, **Transfer Call**, **Deliver Messages**, **Notify Pager**, and **Send Fax** elements can accept a parameter as a number to call.
- The **Send E-mail** element can accept parameters for email addresses.
- The **Text To Speech** element can read parameters to the person called.
- The **Compare Data** element can compare a parameter against several values and route the service based on the outcome.
- The **Create Log Entry** element can write parameters into the log files.

One way to use parameters is to obtain a specific record from a database, as seen in the figure below.

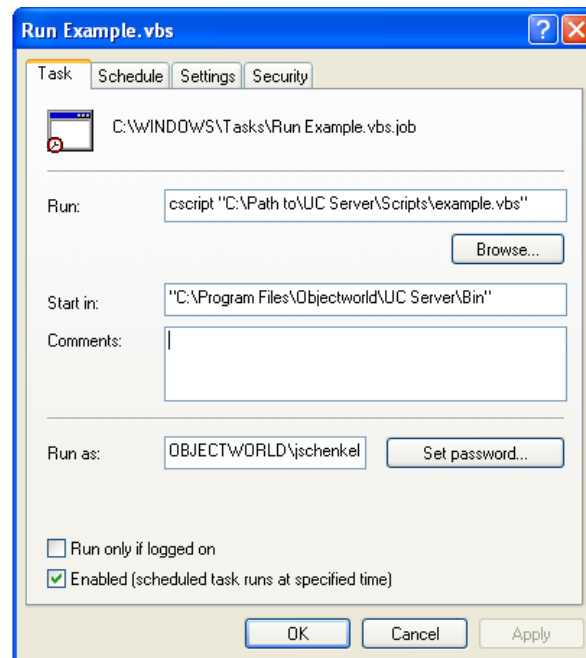


NOTE: The server does not format or interpret the parameters, nor does it ensure that the parameters are in the proper format for the service they are being used in. This is because the server does not know what values the parameters have until the service is executed. Do not mismatch parameters. For example, avoid placing a parameter with text in it into the Transfer Call element.

Configuring the Windows Scheduled Tasks

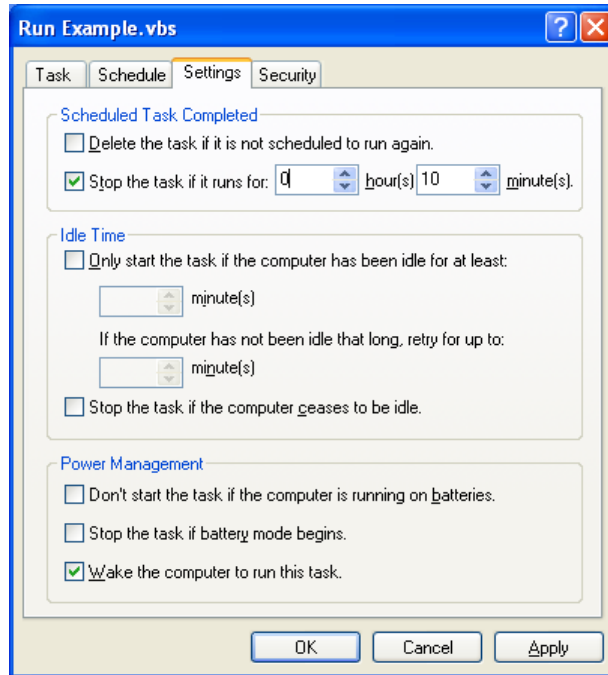
To create a Scheduled Task to run your script:

1. On the **Task** tab, enter **cscript "C:\Path to script\ScriptName.vbs"** in the **Run** field.
2. In the **Start in** field, enter the path to the **.\Bin** folder for the server. Otherwise, the script will not be able to locate Outdial.exe.
3. Enter a user name and password with the appropriate permissions.



4. On the **Schedule** tab, set up the schedule depending on your needs.
5. On the **Settings** tab, select the **Wake the computer to run this task** check box if appropriate.

6. Select the **Stop the task if it runs for** check box and select the time from the drop-down lists. It is recommended that you set the task to stop after 10 minutes, because the scripts typically execute fairly quickly. However, if the database is large, the time should be increased.



Appendix A

Example Scripts

SimpleExample.vbs

```
' =====
' This is a VBScript to automate the use of outdial.exe
' It uses a database for data to queue outbound calls in the UC server.
' It may also be scheduled using Windows Scheduled Tasks.
'
' This script accesses a database, obtains a list of records,
' then runs outdial.exe using data from those records.
'
' Only for use with NetVanta Unified Communications Server 4.2.x or other releases
with outdial.exe.
'
' Usage:
'
' cscript SimpleExample.vbs
'
' =====
' Copyright (c) 2006-2007, Objectworld Communications Corp.
' =====
Option Explicit

'These dim statements create the variables that will be used in the script
'This is necessary because of the "Option Explicit On". That requires all variables
'To be declared with dim before they can be used. This reduces typo errors.
dim strDBLocation, strPhoneField, strTableName
dim strParameter1
dim strUserName, strPassword, strHostName, strServiceLoc
dim objShell, objDB, objCallSheet

'=====Editable Variables====
'=====Database manipulation====

'Full path to Database being used
strDBLocation = "C:\Datasources\Simple.mdb"

'Name of the table being accessed. In this case it's a query table
strTableName = "Query1"

'Name of the field containing the phone number to call
strPhoneField = "Phone"

'Name of the field containing a number unique to each line that will be passed
'as a parameter to the server, which will use it to find that record.
strParameter1 = "ID"

'=====Outdial.exe Parameters====
'Login information
strUserName = "Admin"
strPassword = "password"

'Location of the server. It may only be localhost if run locally on the server.
'IP addresses (192.168.13.37) and URLs (MyServer.MyDomain.MyTLD) will also work
strHostName = "localhost"

'Location of the service that will run after the called party picks up.
```

```

'The path starts in the login's Services folder, so to get simple.srv
'in the Services folder enter a value of simple.srv. To use the Default Trunk Service
'in the Shared folder, enter a value of "\\Shared\Default Trunk Service"
strServiceLoc = "Simple.srv"

'=====

'This creates a shell object so that Outdial may be run from the command line.
Set objShell = CreateObject("WScript.Shell")

'=====Functions=====

'This Function will create an Outdial request to the server.
Function DialOut

'The & _ at the end of three of these lines tell the script interpreter that
'these three lines are to be interpreted as just one line. This is to improve
' readability of the code by limiting it to around 80 characters per line

objShell.exec("C:\Program Files\ADTRAN\NetVanta UC Server\Bin\outdial.exe -u " & _
strUserName & " -pw " & strPassword & " -h " & _
strHostName & " -ph " & objCallSheet(strPhoneField) & _
" -s " & strServiceLoc & " -pm " & objCallSheet(strParameter1))

End Function
'=====

'This creates a database object that will be used to read the database
Set objDB = CreateObject("ADODB." & "Connection")

'This connects the database object objDB to the database in strDBLocation
objDB.Open "Provider =Microsoft.Jet.OLEDB.4.0; Data Source=" & strDBLocation

'This creates a record sheet from all the lines from strTableName
'Because it's a query table, the decision of which lines to select
'Is already made.
Set objCallSheet = objDB.Execute("SELECT * FROM " & strTableName)

'This will loop until all lines in objCallSheet have been dealt with.
'DialOut Calls the function of the same name.
'objCallSheet.MoveNext moves the current record ahead by one.
Do While Not objCallSheet.EOF
    DialOut
    objCallSheet.MoveNext
Loop

'This closes the connection to the database.
objDB.Close

```

NoDatabaseExample.vbs

```

' =====
' This is a VBScript to automate the use of outdial.exe
' It uses a database for data to queue outbound calls in the UC server.
' It may also be scheduled using Windows Scheduled Tasks.
'
' This script accesses a database, obtains a list of records,
' then runs outdial.exe using data from those records.
'
' Only for use with NetVanta Unified Communications Server 4.2.x or other releases
' with outdial.exe.
'
' Usage:

```

```

'
' cscript NoDataBaseExample.vbs
'
' =====
' Copyright (c) 2006-2007, Objectworld Communications Corp.
' =====
Option Explicit

'These dim statements create the variables that will be used in the script
'This is necessary because of the "Option Explicit On". That requires all variables
'To be declared with dim before they can be used. This reduces typo errors.
dim strPhoneNum, strUserName, strPassword, strHostName, strServiceLoc
dim objShell

'=====Editable Variables=====
'Login information
strUserName = "Admin"
strPassword = "password"

'The phone number to call
strPhoneNum = "1333"

'Location of the server. It may only be localhost if run locally on the server.
'IP addresses (192.168.13.37) and URLs (MyServer.MyDomain.MyTLD) will also work
strHostName = "localhost"

'Location of the service that will run after the called party picks up.
'The path starts in the login's Services folder, so to get HelloWorld.srv
'in the Services folder enter a value of HelloWorld.srv. To use the Default Trunk
Service
'in the Shared folder, enter a value of "\\Shared\Default Trunk Service"

strServiceLoc = "\\Shared\Default Trunk Service.srv"

'=====

'This creates a shell object so that outdial may be run from the command line.
Set objShell = CreateObject("WScript.Shell")

'=====Functions=====

'This Function will create an outdial request to the server.
Function DialOut

'The & _ at the end of three of these lines tell the script interpreter that
'these three lines are to be interpreted as just one line. This is to improve
' readability of the code by limiting it to around 80 characters per line

objShell.Run("outdial -u " & _
strUserName & " -pw " & strPassword & " -h " & _
strHostName & " -ph " & strPhoneNum & " -s " & strServiceLoc)

End Function

'=====

'This runs the DialOut Function
DialOut

```