



Open Queue v2.0

Swyx Forum

<http://projects.swyx-forum.com>

1	License	4
2	Overview.....	5
2.1	Features.....	5
2.2	Differences Open Queue / SwyxWare Queue.....	5
2.3	Requirements	6
2.4	Updates	6
2.5	Support	6
3	Setup	7
3.1	Scripts	7
3.1.1	On SwyxWare v6.02.....	7
3.1.2	On SwyxWare v6.1x (or higher)	8
3.2	Database.....	11
3.3	Webserver.....	26
3.4	Updates	33
3.4.1	Update from v2.0 Beta 1 to RC 1	33
3.4.2	Update from v2.0 RC 1 to RC 2	33
3.4.3	Update from v2.0 RC 2 to RC 3	33
3.4.4	Update from v2.0 RC 3 to final Release.....	34
3.5	Longest Waiting	35
3.5.1	Setup Call Detail Records.....	35
3.5.2	Setup Longest Waiting Table and Trigger.....	35
3.5.3	Setup Longest Waiting Scripts	35
4	Usage.....	36
4.1	How to use Open Queue actions	36
4.2	Simple Call Routing Script	38
4.3	Queue Visualization.....	39
4.3.1	Functions	40
4.3.2	URL Parameter	41
4.3.3	Functionality Configuration	41
5	GSE Actions.....	42
5.1	OpenQueue.CreateQueue	42
5.1.1	Parameter	42
5.1.2	Return Values	43
5.2	OpenQueue.DeleteQueue	43
5.2.1	Parameter	43
5.2.2	Return Values	43
5.3	OpenQueue.CloseQueue.....	44
5.3.1	Return Values	44
5.4	OpenQueue.SetParameter.....	45
5.4.1	Parameter	45
5.4.2	Return Values	48

5.5	OpenQueue.GetParameter.....	49
5.5.1	Parameter	49
5.5.2	Return Values	50
5.6	OpenQueue.ProcessCallByQueue	51
5.6.1	Parameter	52
5.6.2	Return Values	53
5.7	OpenQueue.RemoveCallFromQueue.....	54
5.7.1	Parameter	54
5.7.2	Return Values	54
5.8	OpenQueue.WaitForDisconnect.....	55
5.8.1	Parameter	55
5.8.2	Return Values	55
6	Reference	56
6.1	Programming model	56
6.2	Business Logic	57
6.2.1	COpenQueue Class.....	57
6.3	Provider	58
6.3.1	COpenQueue_GenericDBProvider Class	58
6.3.2	Database	59
6.3.3	OpenQueue_Calls Table	59
6.3.4	OpenQueue_Status Table	59
6.3.5	IpPbxCDR	60
6.3.6	LongestWaiting Table	60

1 License

Open Queue Project v2.0

This is a Swyx Forum Open Source Project.
Read more about it at <http://projects.swyx-forum.com>

Copyright (c) 2007-2008 by Swyx Forum
Copyright (c) 2007-2008 by Tom Wellige
All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice must be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2 Overview

This project provides call queuing functionality. It is derived from the Swyx Knowledgebase Article kb2356 explaining how to implement a call queue and extends it with lots of new functionality and robustness.

Taking the kb article as version v1.0 the Open Queue project starts with v2.0.

2.1 Features

- handle any number of different call queues in different ecr scripts
- keeps a call on hold (playing music on hold) until destination is reachable
- free definable queue size
- free definable queue timeout
- announce "please wait" before and during music on hold
- announce position within queue during music on hold
- use destination group's configured agent selection (hunt groups) or longest waiting
- store user defined data per call (e.g. a previously entered ticket id)
- complete queue visualization (asp webpage), incl. the following features:
 - list all calls within a queue incl. caller number, caller name, connected name, start date, duration
 - pick a call from the queue
 - cancel (disconnect) a call from the queue
 - move a call to the top of the queue
 - close queue, meaning the queue will not accept new calls
 - cancel all calls within the queue

2.2 Differences Open Queue / SwyxWare Queue

- visualization of a queue
- pick a call from the queue
- cancel (disconnect) call from the queue
- alter position within queue, meaning boost call to top position
- close queue
- cancel (disconnect) all calls from the queue
- use destination group's configured agent selection (hunt groups) or longest waiting
- store userdefined data per call (e.g. a previously entered ticket id)
- try to connect a call first before starting music on hold
- free defineable connect timeout
- free defineable connect alert sound

2.3 Requirements

- SwyxWare / NetPhone v6.02 or newer
- SwyxECR or OptionPack Professional Licenses
- MS SQL Server 2005/2005 or MSDE (it is possible to use the MSDE the SwyxWare uses itself)
- MS Internet Information Server IIS (for visualization)
- When using the "Longest Waiting" feature the SwyxServer must write the Call Detail Records (CDR) into the database (same as OpenQueue uses) instead of a text file. See manual for details.

2.4 Updates

This project will be updated in irregular periods of time. Check the project's home page from time to time or subscribe to the project's Blog RSS feed:

<http://www.swyx-forum.com/community/Projects/OpenQueue/tabid/70/Default.aspx>

To subscribe to the RSS feed click on the  symbol on top of the Blog area on that page. Current available downloads can be found at the project's download page:

<http://www.swyx-forum.com/community/Projects/OpenQueue/Downloads/tabid/72/Default.aspx>

2.5 Support

This is a Swyx Forum open source project. All support, enhancement and bug requests must be placed into the project's forum at

<http://www.swyx-forum.com/community/Projects/OpenQueue/Forums/tabid/71/Default.aspx>

No other support is available!

Use the "Using Open Queue" forum for all usage related topics, use the "Development" forum for all development, enhancement related topics.

Additionally you can check the project's issue tracker if a problem you want to report is already known:

<http://defects.swyx-forum.com/?target=http://www.codeplex.com/openqueue/WorkItem/List.aspx>

In regard of the license being provided to you (see chapter 1) there is neither a guaranteed response time for support requests nor a guaranteed solution.

3 Setup

Download the latest version of Open Queue. The download file is a ZIP file. When extracting it, make sure it's subfolder structure is extracted correctly.

3.1 Scripts

3.1.1 On SwyxWare v6.02

You need to copy a number of files to the SwyxWare server machine.

If you want to have the Open Queue functionality available for a certain user the target folder of the following instruction is

```
C:\Documents and Settings\All Users\  
Application Data\Swyx\Share\User\nnn\PhoneClient\Scripts
```

where **nnn** is the SwyxWare Username.

If you want to have the Open Queue functionality available for all users the target folder of the following instruction is

```
C:\Documents and Settings\All Users\  
Application Data\Swyx\Share\Data\PhoneClient\Scripts
```

If your SwyxWare server machine is not an English system, the “Documents and Settings” folder might be named localized, e.g. “Dokumente und Einstellungen” if you are using a German server machine.

If you are using an OEM version of SwyxWare, the folder below “Application Data” is named differently, e.g. “T-Com” for Octopus NetPhone Applikationslösung.

Copy all files from the **ase** folder of the install package into the target folder.

```
actionLongestWaiting.ase  
actionLongestWaiting.vbs  
actionOpenQueueCloseQueue.ase  
actionOpenQueueCloseQueue.vbs  
actionOpenQueueCreateQueue.ase  
actionOpenQueueCreateQueue.vbs  
actionOpenQueueDeleteQueue.ase  
actionOpenQueueDeleteQueue.vbs  
actionOpenQueueGetParameter.ase  
actionOpenQueueGetParameter.vbs  
actionOpenQueueProcessCallByQueue.ase  
actionOpenQueueProcessCallByQueue.vbs  
actionOpenQueueRemoveCallFromQueue.ase  
actionOpenQueueRemoveCallFromQueue.vbs  
actionOpenQueueSetParameter.ase  
actionOpenQueueSetParameter.vbs  
OpenQueue_GenericDBProvider.vbs
```

3.1.2 On SwyxWare v6.1x (or higher)

From SwyxWare v6.10 on all relevant files are kept within SwyxWare's IPPBX database. So all Open Queue script files have to be uploaded into this database too.

To do so you can either use the **SwyxWare Administration** or an additional tool from the SwyxWare CD, call **SwyxWare File Explorer**.

The following files from the `ase` folder of the install package need to be uploaded to the database:

```
actionLongestWaiting.ase  
actionLongestWaiting.vbs  
actionOpenQueueCloseQueue.ase  
actionOpenQueueCloseQueue.vbs  
actionOpenQueueCreateQueue.ase  
actionOpenQueueCreateQueue.vbs  
actionOpenQueueDeleteQueue.ase  
actionOpenQueueDeleteQueue.vbs  
actionOpenQueueGetParameter.ase  
actionOpenQueueGetParameter.vbs  
actionOpenQueueProcessCallByQueue.ase  
actionOpenQueueProcessCallByQueue.vbs  
actionOpenQueueRemoveCallFromQueue.ase  
actionOpenQueueRemoveCallFromQueue.vbs  
actionOpenQueueSetParameter.ase  
actionOpenQueueSetParameter.vbs  
OpenQueue_GenericDBProvider.vbs
```

If Open Queue should be available for a single user only:

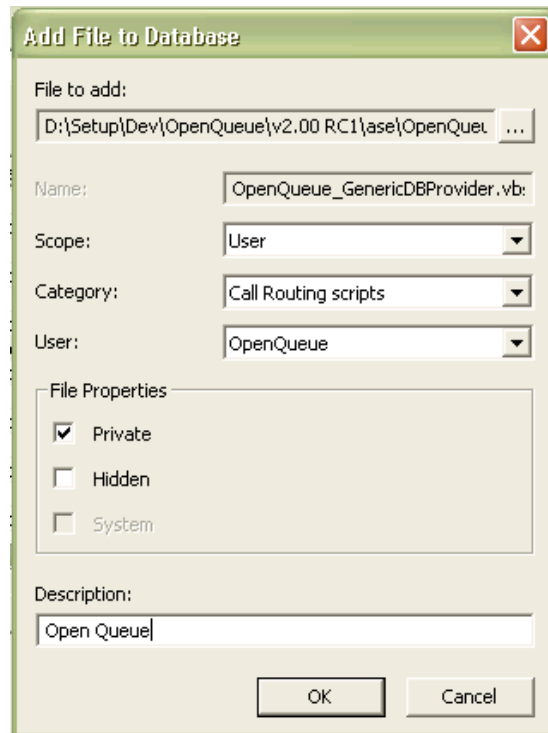
Scope:	User
Category:	Call Routing Scripts
User:	Select the User Name
File Properties:	Private

If Open Queue should be available for all users:

Scope:	Global
Category:	Call Routing Scripts
File Properties:	

Usage of SwyxWare Administration to upload script files

1. Open **SwyxWare Administration**
2. Open **Server Properties**
3. Open the **Files** tab
4. Click on **Edit...**
5. Click on **Add...**



The screenshot shows a dialog box titled "Add File to Database" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- File to add:** A text box containing the path "D:\Setup\Dev\OpenQueue\v2.00 RC1\ase\OpenQueue" followed by a browse button "...".
- Name:** A text box containing "OpenQueue_GenericDBProvider.vb".
- Scope:** A dropdown menu with "User" selected.
- Category:** A dropdown menu with "Call Routing scripts" selected.
- User:** A dropdown menu with "OpenQueue" selected.
- File Properties:** A section with three checkboxes: "Private" (checked), "Hidden" (unchecked), and "System" (unchecked).
- Description:** A text box containing "Open Queue".
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

6. Select **all files** from the **ase** folder
7. Select **Scope**
8. Select **Category**
9. Select **User**
10. Set the **File Properties**
11. Enter **Description**
12. Click **OK**

3.1.2.1 Usage of SwyxWare File Explorer to upload script files

1. **Install** the **SwyxWare File Explorer** from the SwyxWare CD\Tools\SwyxWare File Explorer folder
2. **Start** SwyxWare File Explorer
3. **Connect** to the SwyxWare Database
4. Click on **Upload** in the Toolbar

Upload File

File(s) to upload:
(multiple) Select...

Name: (multiple)

Scope: User

Category: CallRoutingScripts

Users: OpenQueue Select...

☒ File is private
☐ File is hidden
☐ System file
☒ Compressed

Description:
Open Queue

OK Cancel

5. Select **all files** from the **ase** folder
6. Select **Scope**
7. Select **Category**
8. Select **User**
9. Set the **File Properties**
10. Enter a **Description**
11. Click **OK**

3.2 Database

Open Queue v2.0 comes with a generic database provider making use of Microsoft's ADO database functionality. Therefore all databases being available for Microsoft ADO, i.e. coming with an OLEDB Provider, are useable for Open Queue.

Most likely you will decide to use MSDE.

There isn't a complete database setup currently available (this might change in the future). You need to create a new database, configure user rights and run SQL scripts to install Open Queue properly. This could all be done via OSQL commands, but for reason of comfort the following instruction is based on the **Microsoft SQL Server Management Studio Express**, which is available for free from Microsoft and manages SQL Server 2005 databases as also "old" MS SQL Server 2000/MSDE databases.

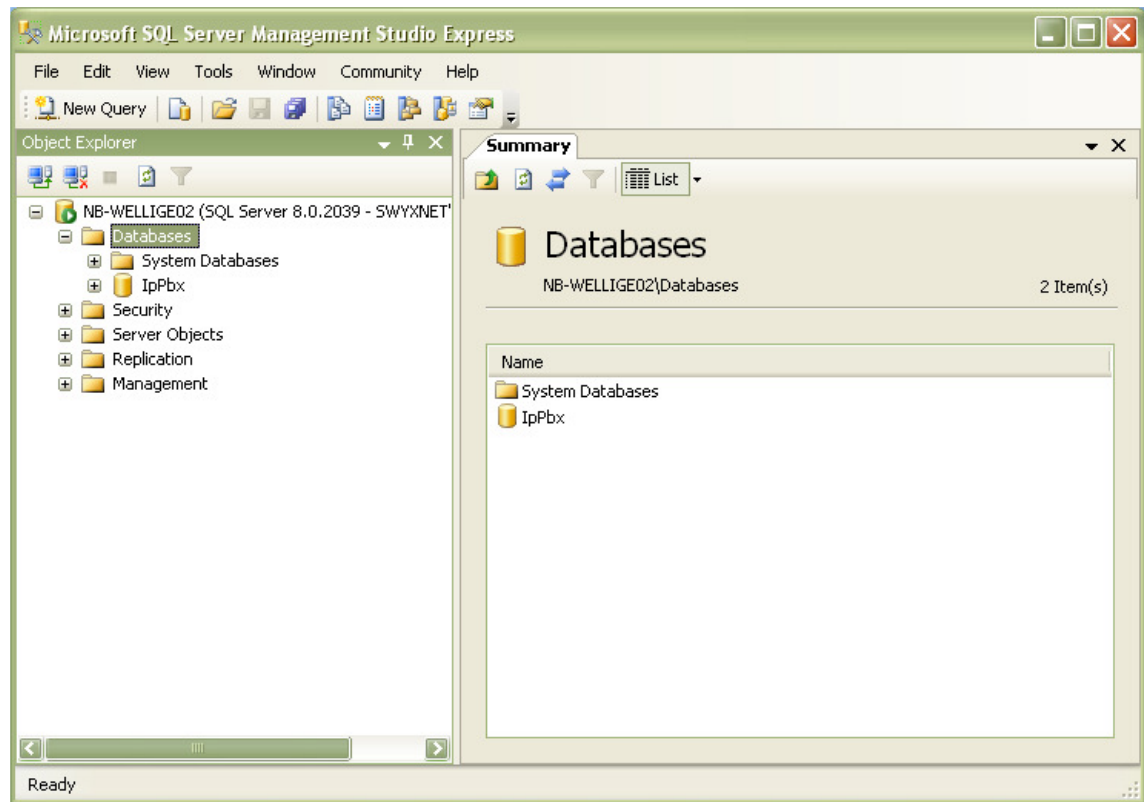
You can get your own copy of MS SSMSE from the following URL:

<http://www.microsoft.com/sql/downloads/2005/default.msp>

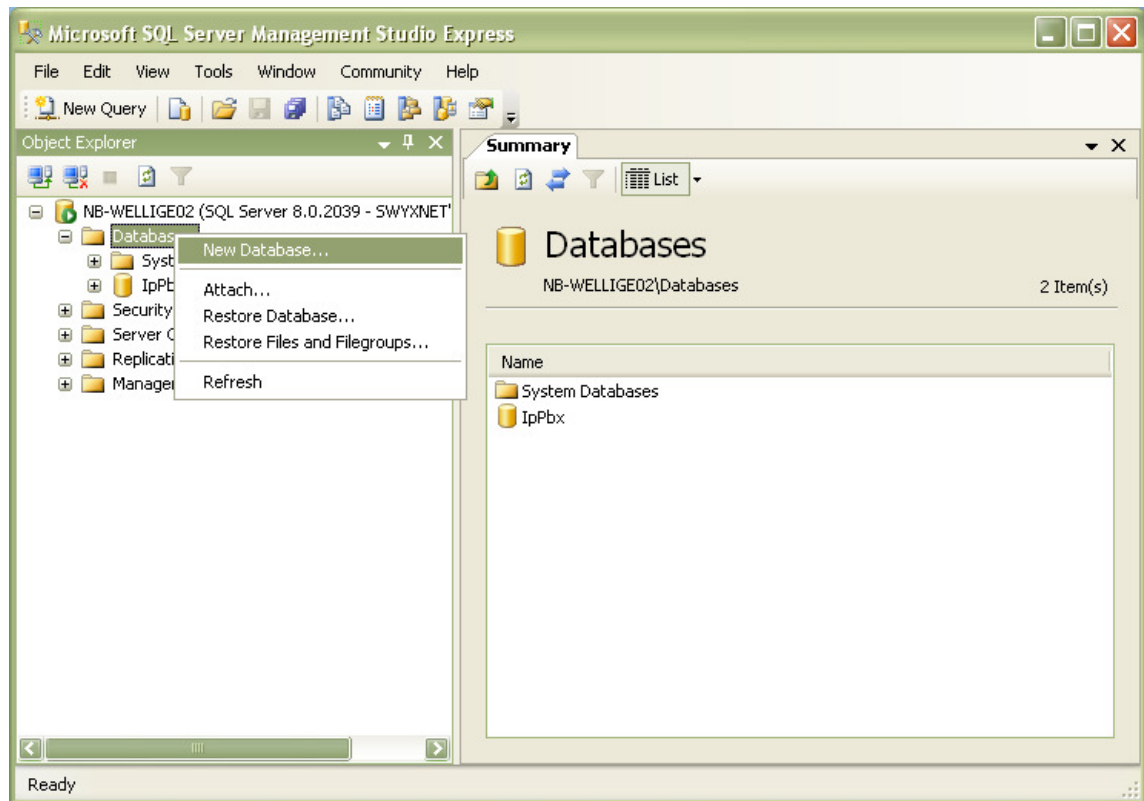
The following instruction installs the Open Queue database into the MSDE which is installed on the SwyxWare server machine and is already used by the SwyxWare itself.

If you want to make use of the **Longest Waiting** feature read **chapter 3.4** first before following the next steps.

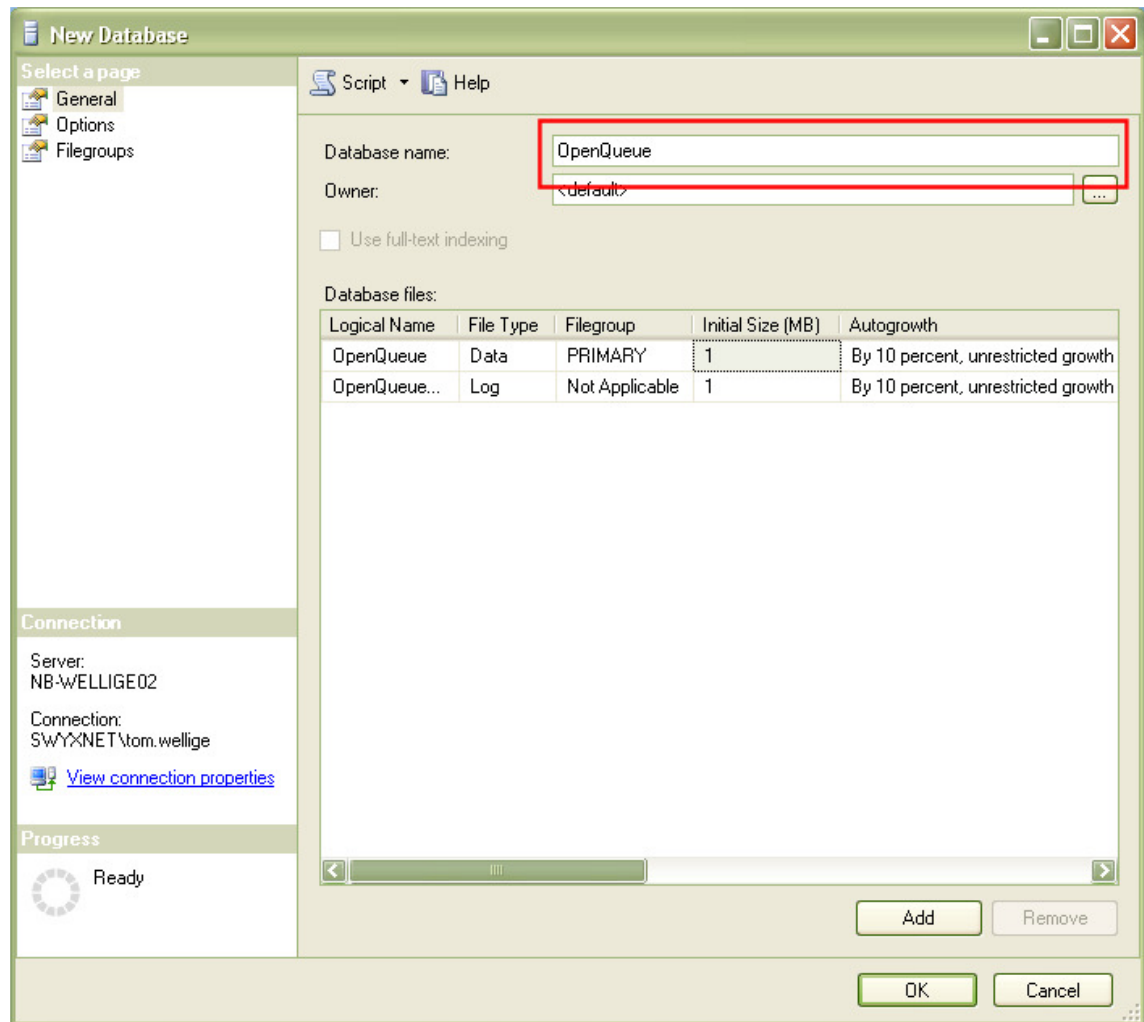
1. Open MS SSMSE



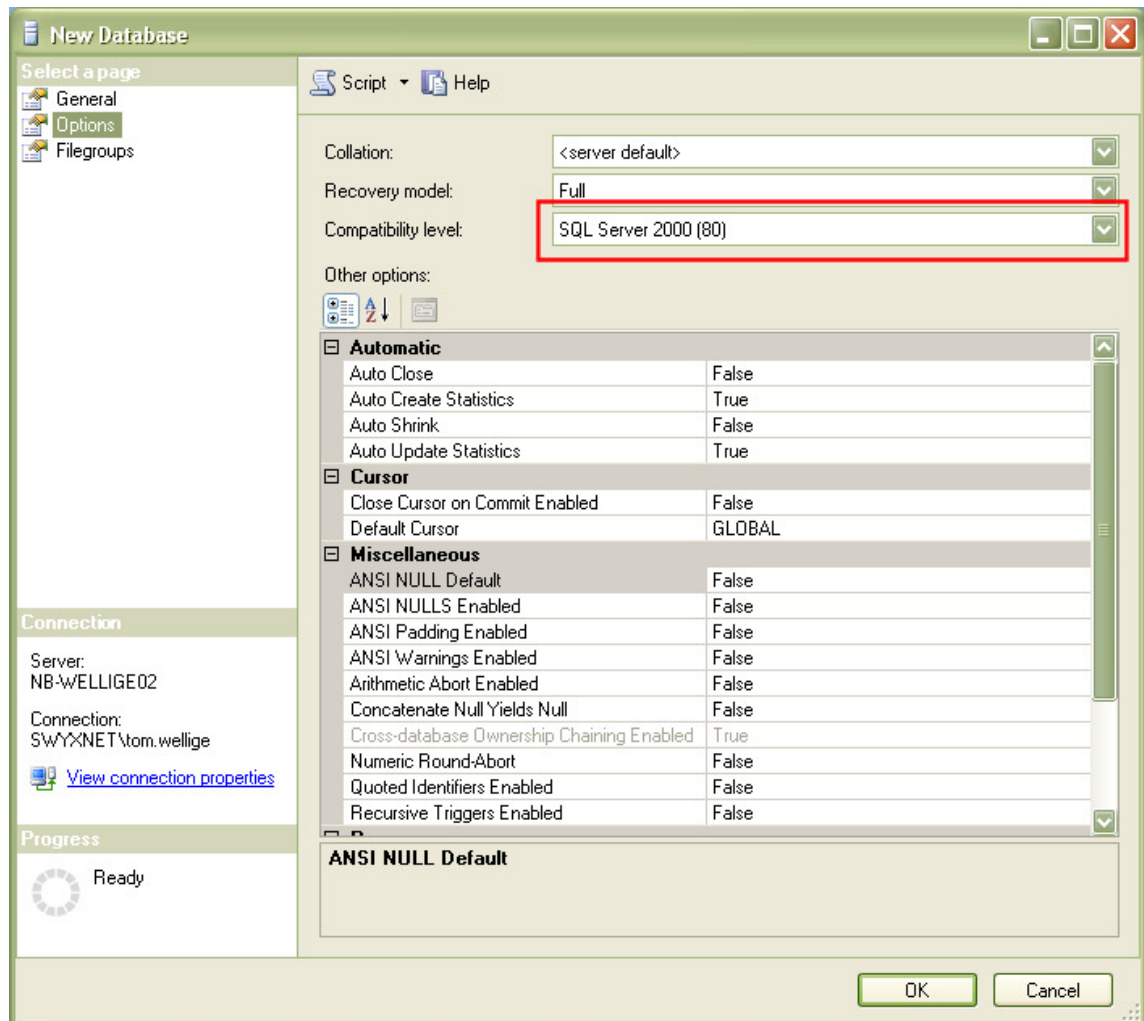
2. Right click **Databases** and select **New Database...** to create a new database.



3. Enter the name of the new Database, **OpenQueue** on the **General** page.

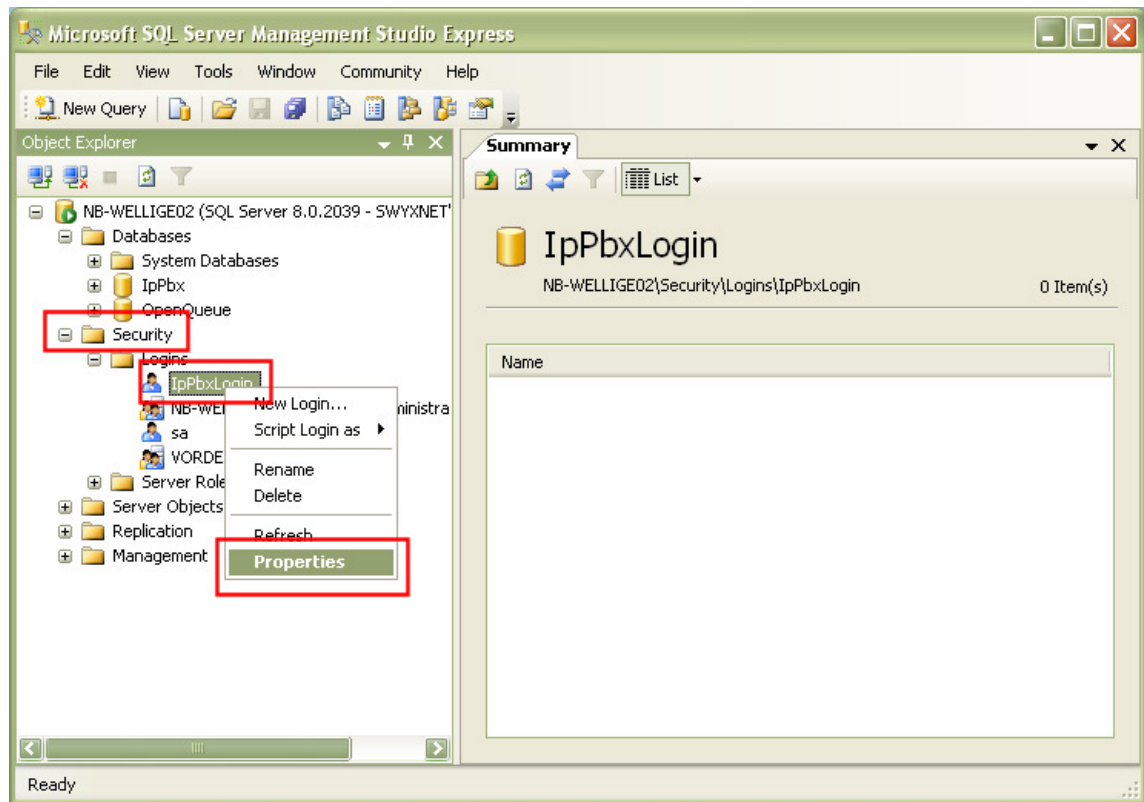


4. Just for clarification check the **Options** page if the compatibility level is set to “SQL Server 2000”.

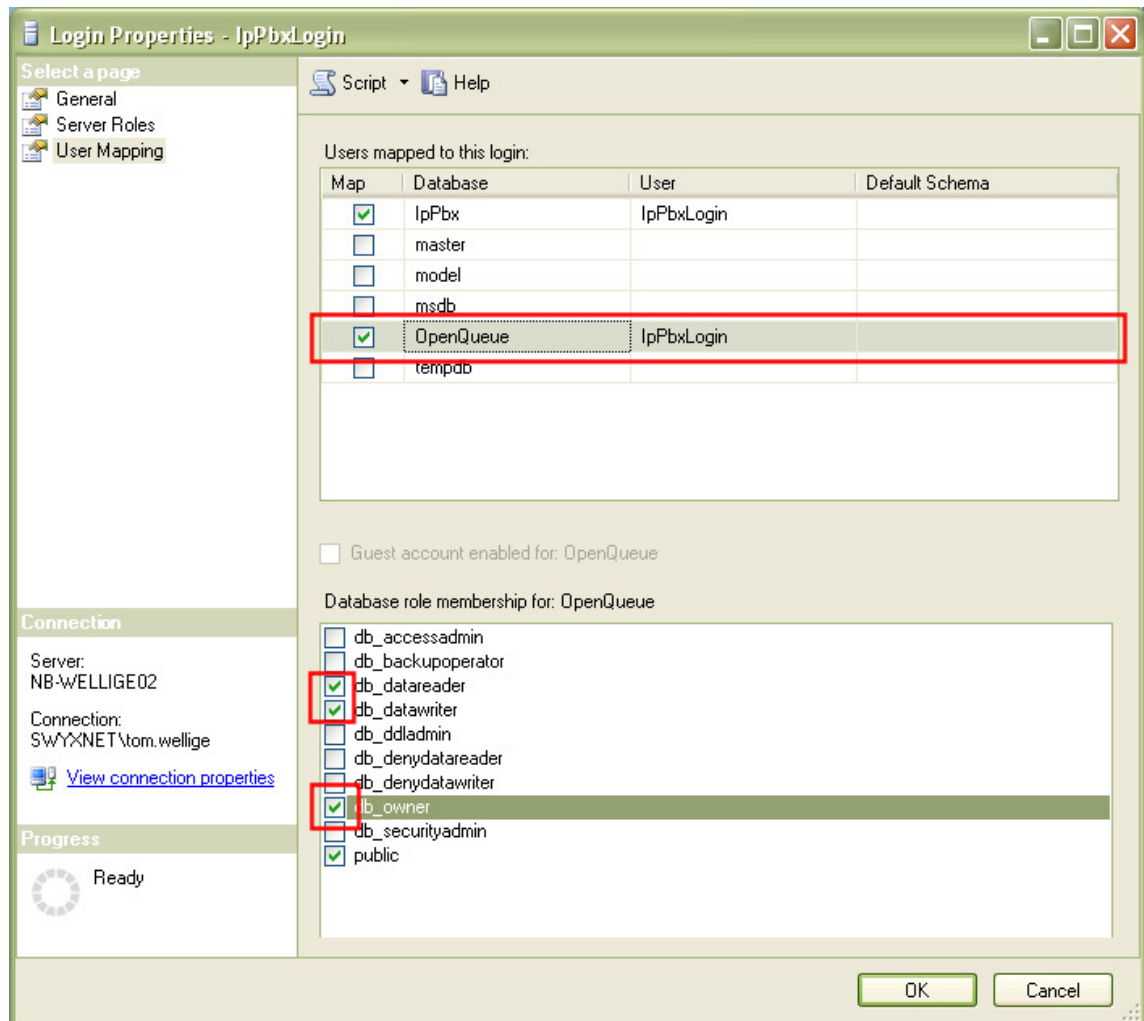


5. Click on **OK** to create the new database.
6. Within the next steps you need to configure database logins and access rights for SwyxWare Server (who runs the ECR scripts) and the Web Server (if you want to make use of the visualization which is explained in the next chapter).

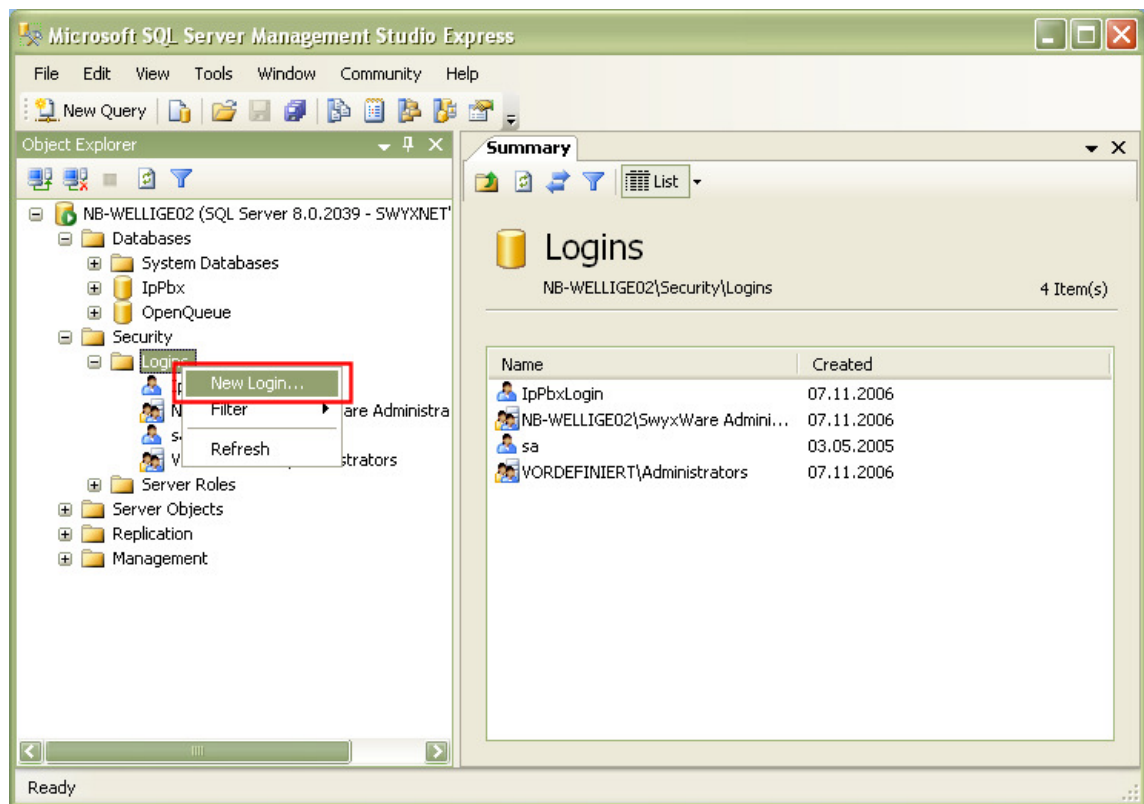
7. First step is to give the IpPbxLogin access to the new database. To do so right click **IpPbxLogin** within **Security / Logins** node and select **Properties**



8. Tick the checkbox **OpenQueue** in the first list and **db_datareader**, **db_datawriter** and **db_owner** in the second list on the **User Mapping** page.



9. If you want to make use of the visualization you need to create a new login for the web server service (this requires the IIS (Internet Information Server) to be installed on the SwyxWare server machine). To do so right click on **Logins** and select **New Login...**

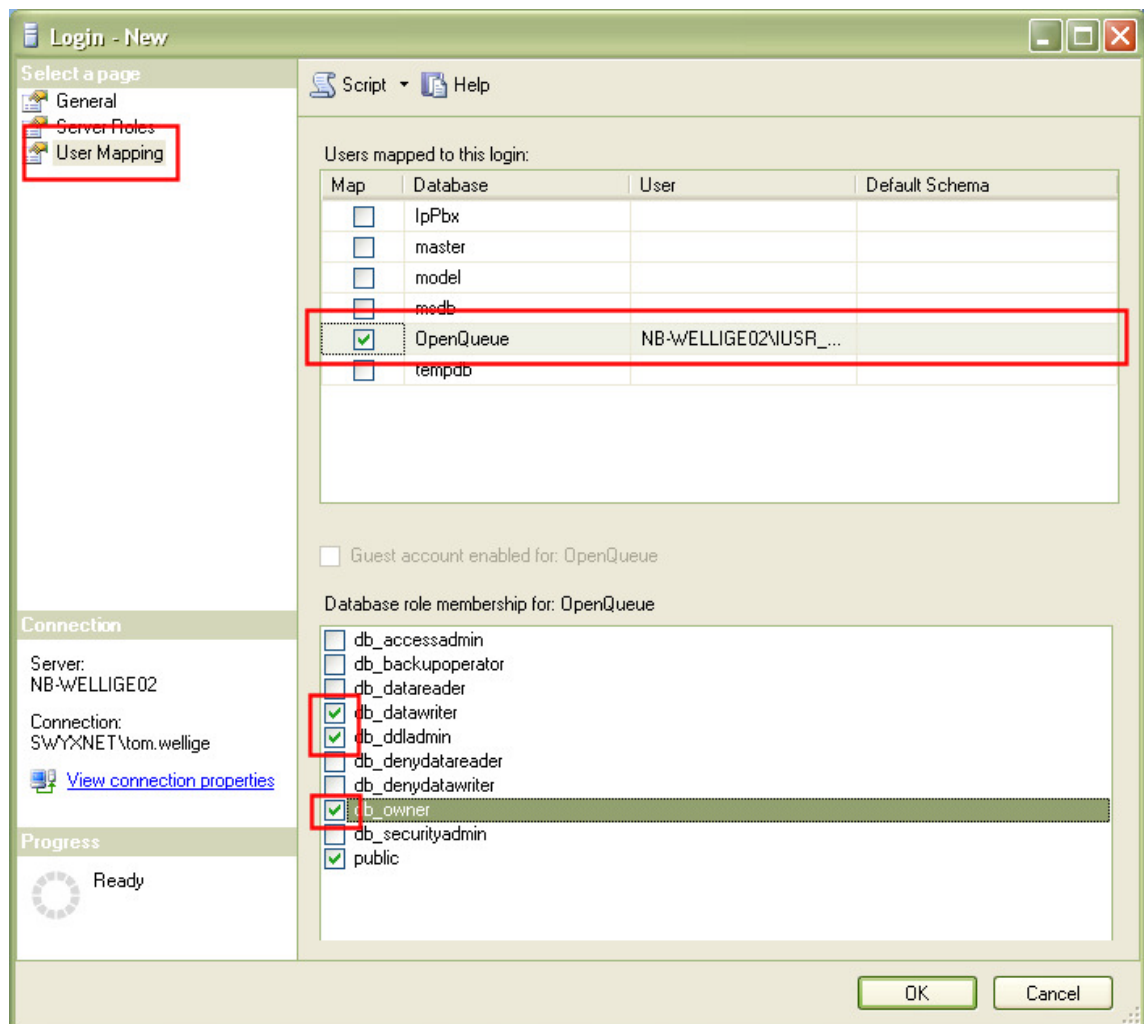


10. Select the local **IUSER_XXX** user from your server machine and set the **Default Database** to **OpenQueue** on the **General** page.

The screenshot shows the 'Login - New' dialog box with the following details:

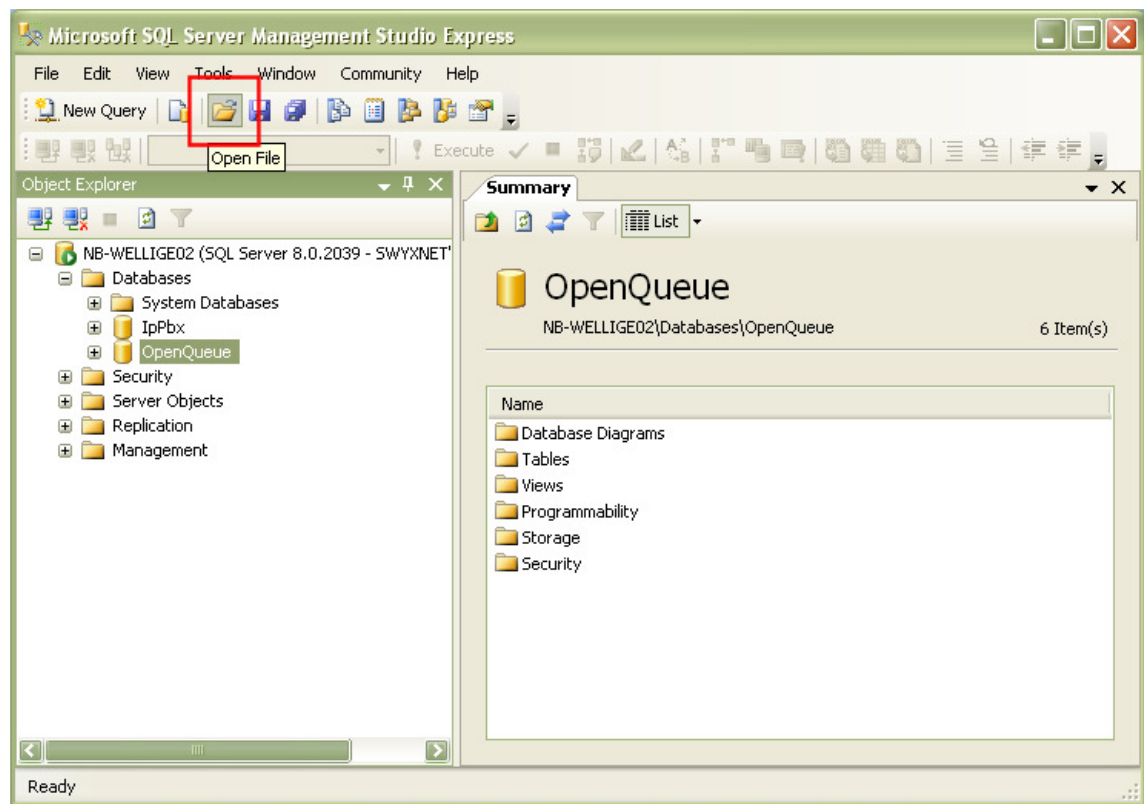
- Left Pane:** 'General' is selected under 'Select a page'. Below it, 'Connection' shows 'Server: NB-WELLIGE02' and 'Connection: SWYXNET\Tom.wellige'. 'Progress' shows 'Ready'.
- Main Area:**
 - Login name:** NB-WELLIGE02\IUSR_NB-WELLIGE02 (highlighted with a red box).
 - Authentication:** 'Windows authentication' is selected.
 - Default database:** OpenQueue (highlighted with a red box).
 - Default language:** <default> (highlighted with a red box).
- Buttons:** 'OK' and 'Cancel' are at the bottom right.

11. To give this new user access to the OpenQueue database change to the **User Mapping** page and tick the following checkboxes: **OpenQueue**, **db_datareader**, **db_datawriter** and **db_owner**.

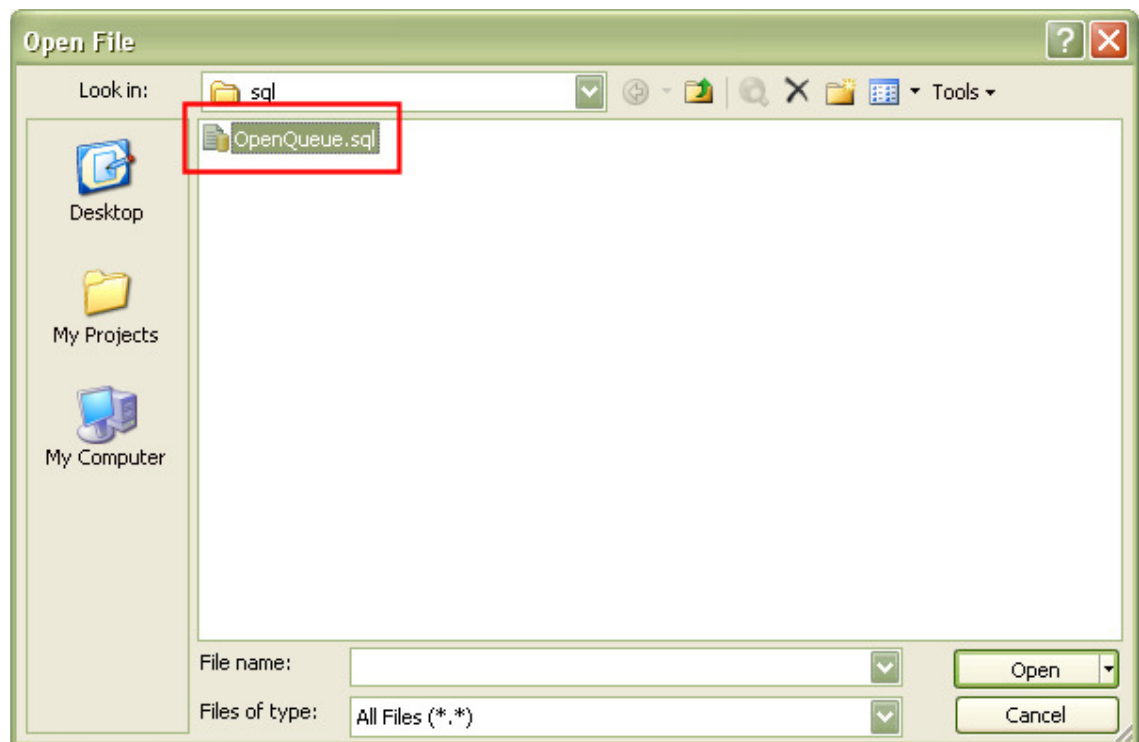


12. So, the database is created and the necessary user logins are configured. Now you need to create the needed tables and insert data by using an existing sql script.

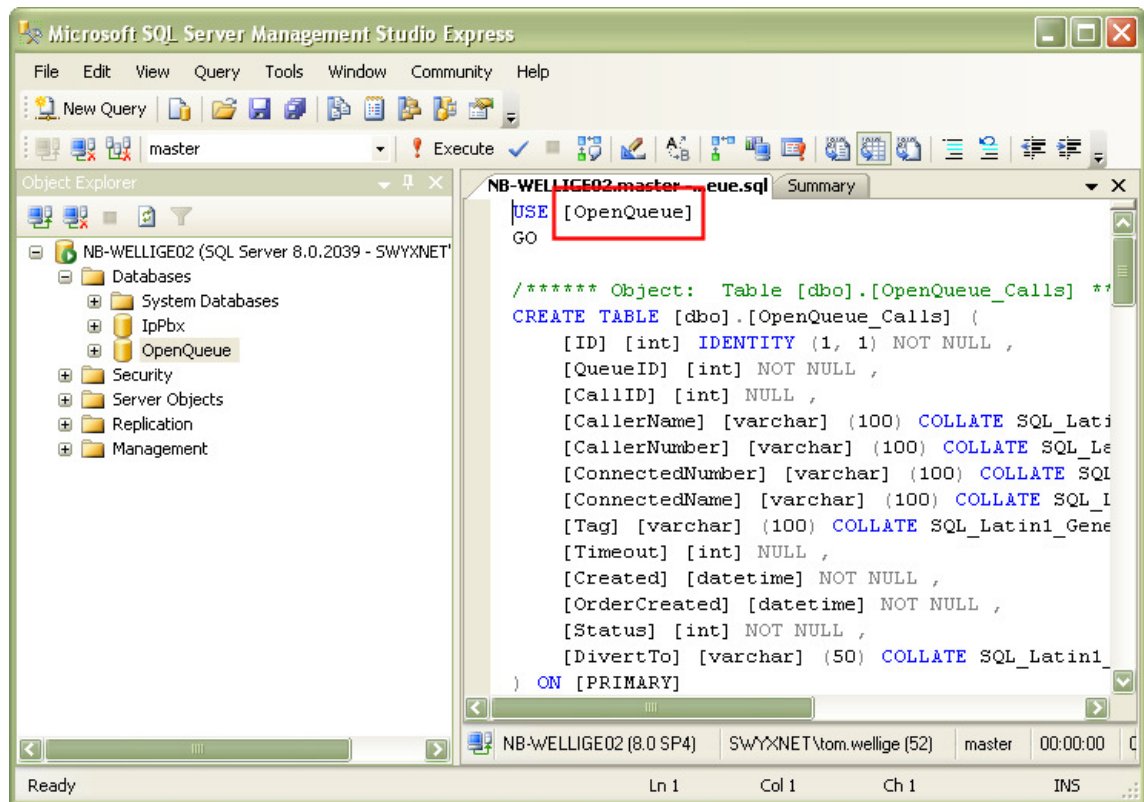
13. Click on the **Open File** icon within the toolbar.



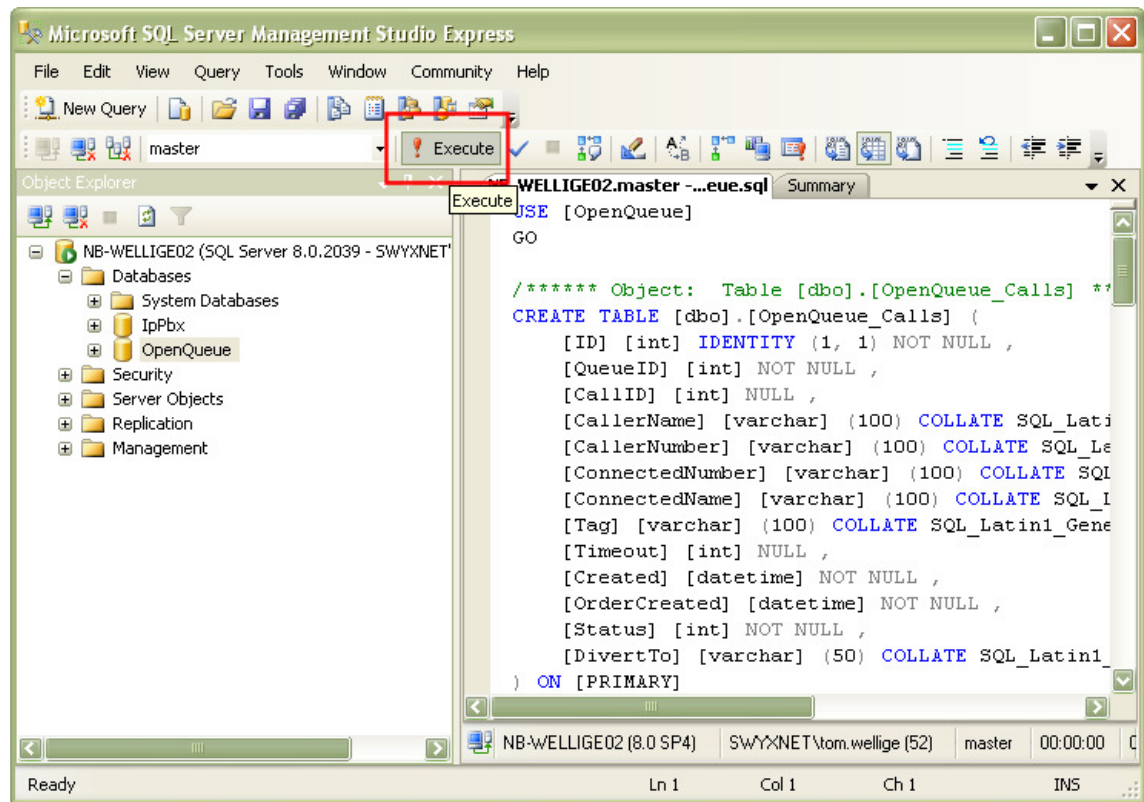
14. Select the **OpenQueue_1.0.2.sql** file. (the version number might vary to the current version)



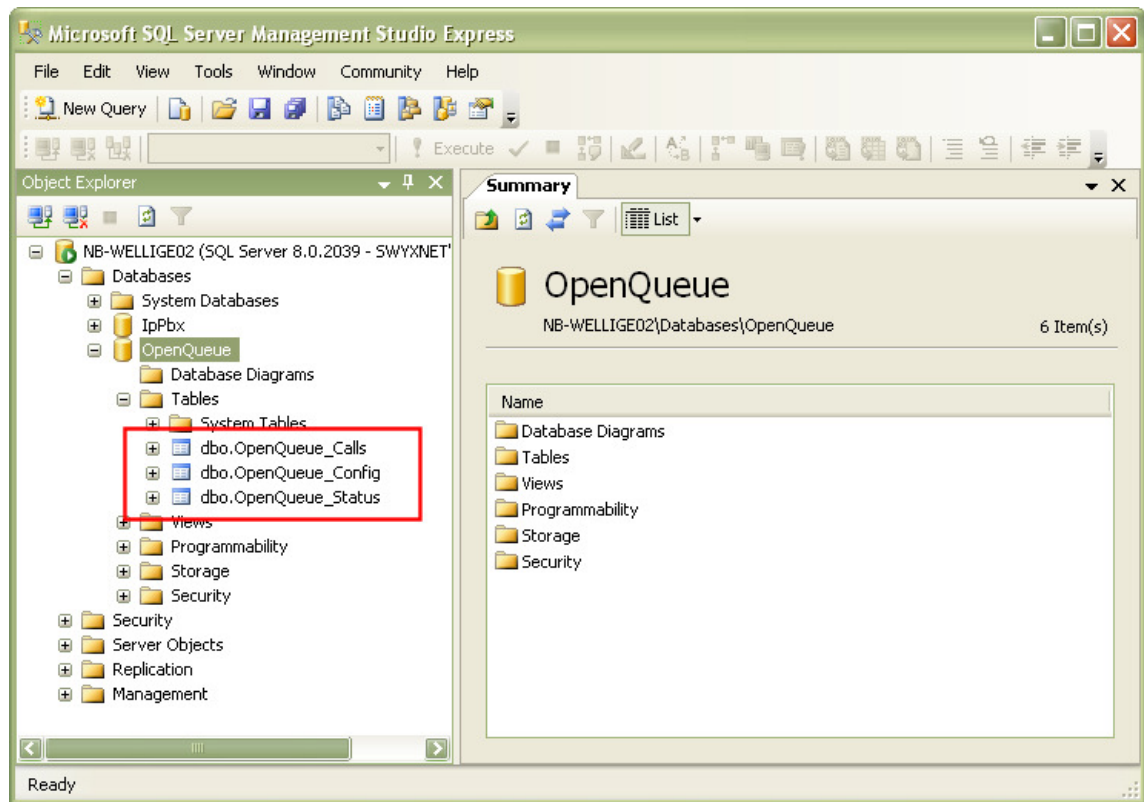
15. If you have renamed the database to something else than **OpenQueue** you need to modify the first line of the opened script accordingly. This is most likely the case if you use the Longest Waiting feature and use the **Call Detail Records** database.



16. Click on the **Execute** button within the toolbar.



17. Check the results by closing the script window (click the X in the top right corner) and press **F5** to refresh the view.



18. **Congratulations!** Your database is now up and running.

Some final words on the database connection. The above instruction configures existing Windows User accounts to get access to the Open Queue database. With this Windows Security can be used within the ECR scripts and Web page script to connect to it. You will not be forced to enter any login information into the scripts. Of course you can configure any other database login and enter the login information into the scripts. But this is not recommended.

3.3 Webserver

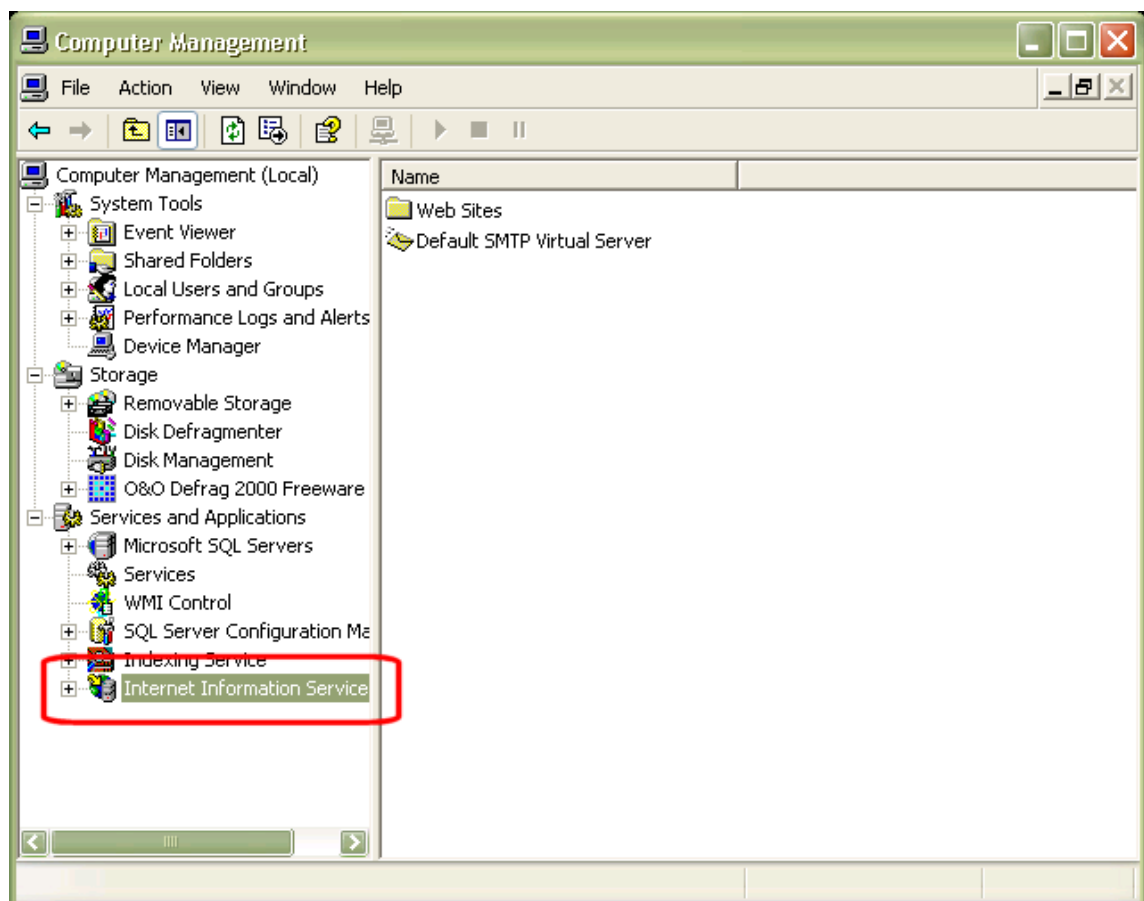
If you want to use the queue visualization coming with Open Queue you need to have a Microsoft Internet Information Server (IIS). This can either be installed/running on the SwyxWare server machine or any other Windows machine (from Windows XP up to Windows Server 2003) in your domain. The following instruction assumes you are using the IIS on the SwyxWare server machine.

If you haven't installed the IIS yet do it now via

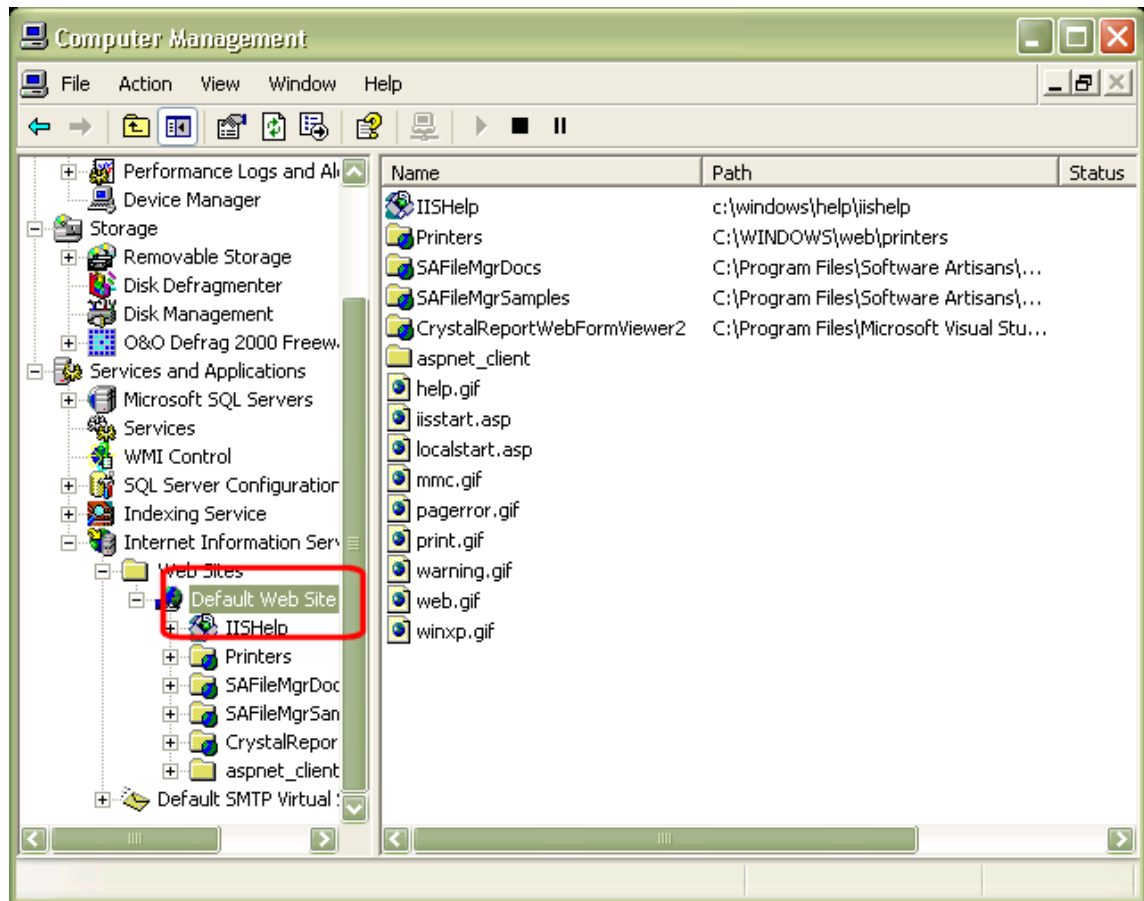
- Control Panel
- Add or Remove Programs
- Add/Remove Windows Components
- Internet Information Services (IIS)

Once this is done follow the next steps to configure the local IIS to run the Open Queue web visualization as default web page.

1. Open the **Computer Management** and open the **Services and Applications** node.

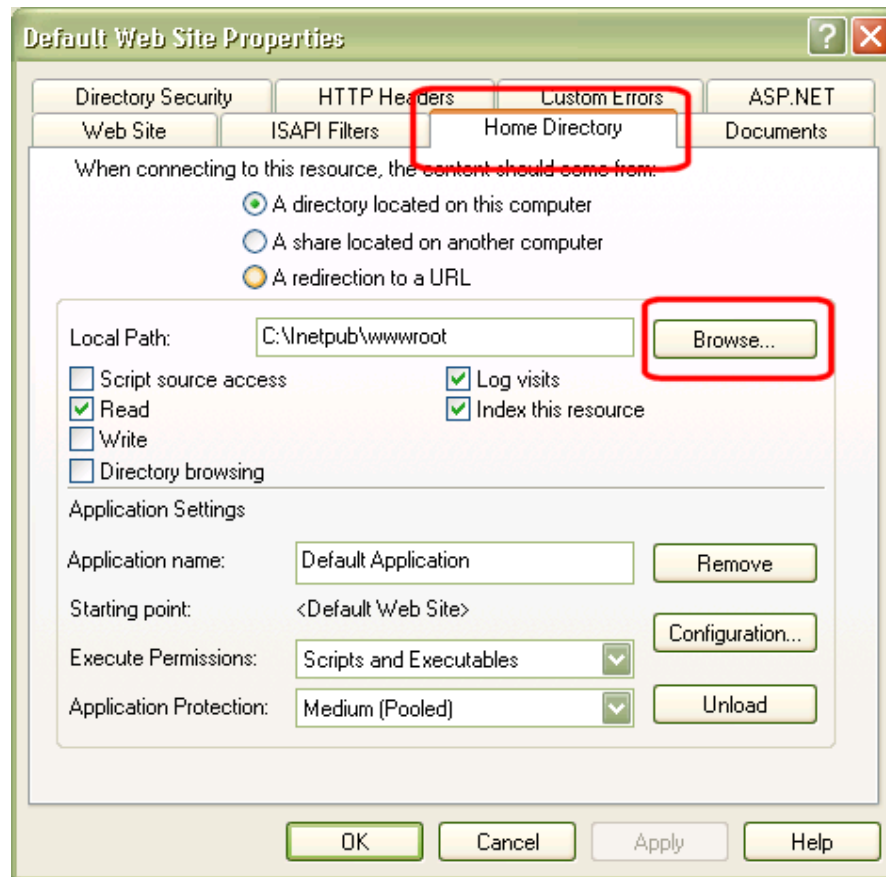


2. Open the **Internet Information Service** node.



The current default web site should be something like the above, i.e. C:\inetpub\wwwroot.

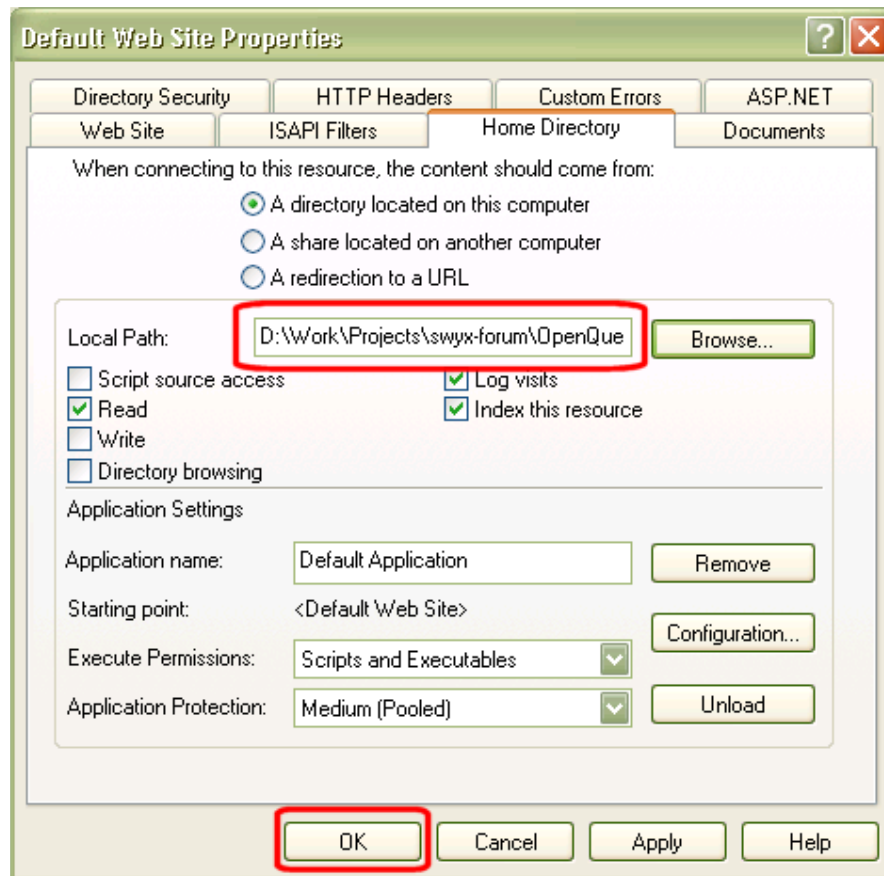
3. Open the **Properties** of the **Default Web Site** by right clicking it and switch to the Home Directory tab.



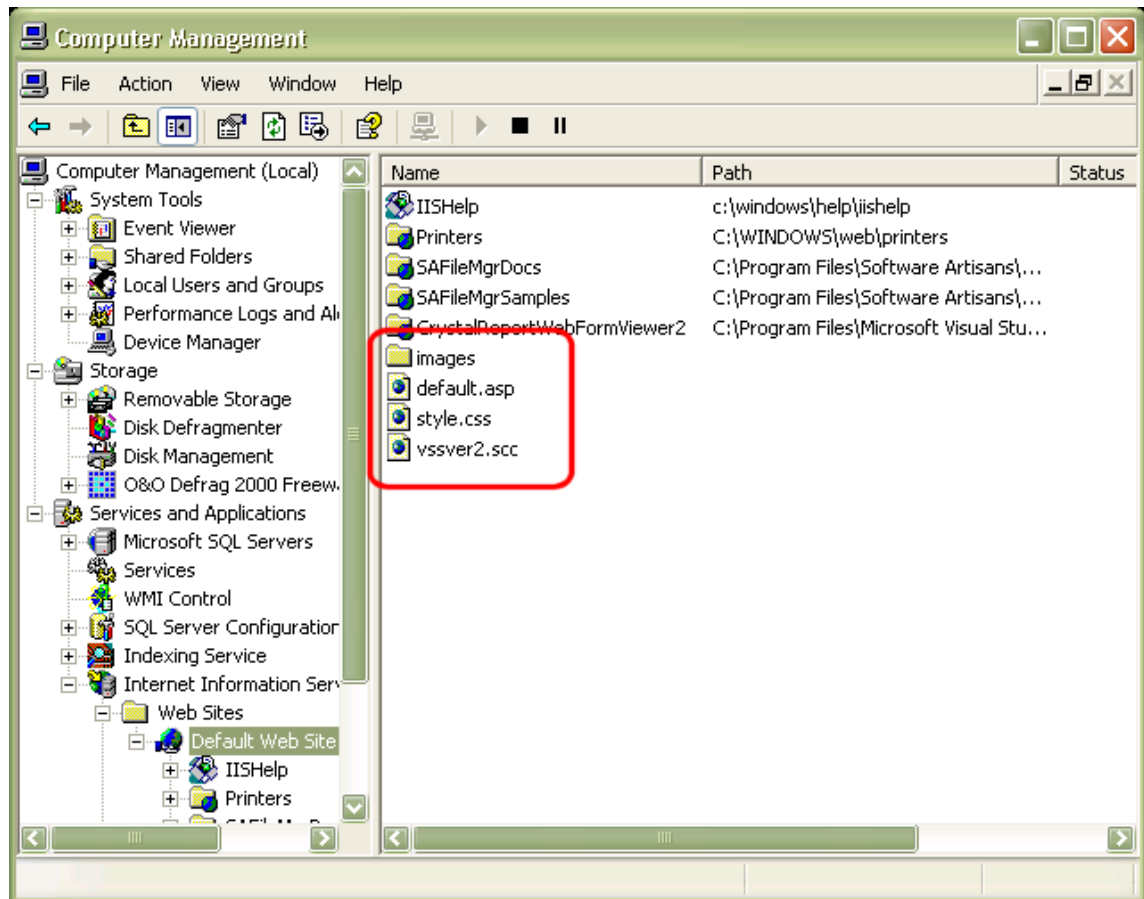
4. Click on **Browse** to select the folder where you have extracted the Open Queue package to, in detail, select the **web** subfolder.



5. Click on **OK** to store the settings.



6. Check the results within the **Computer Management**.



7. **IMPORTANT:** Open the file **default.asp** using a text editor and update the line 31, the database connect string. You have to update the name of the machine the Open Queue database is installed on instead of localhost:

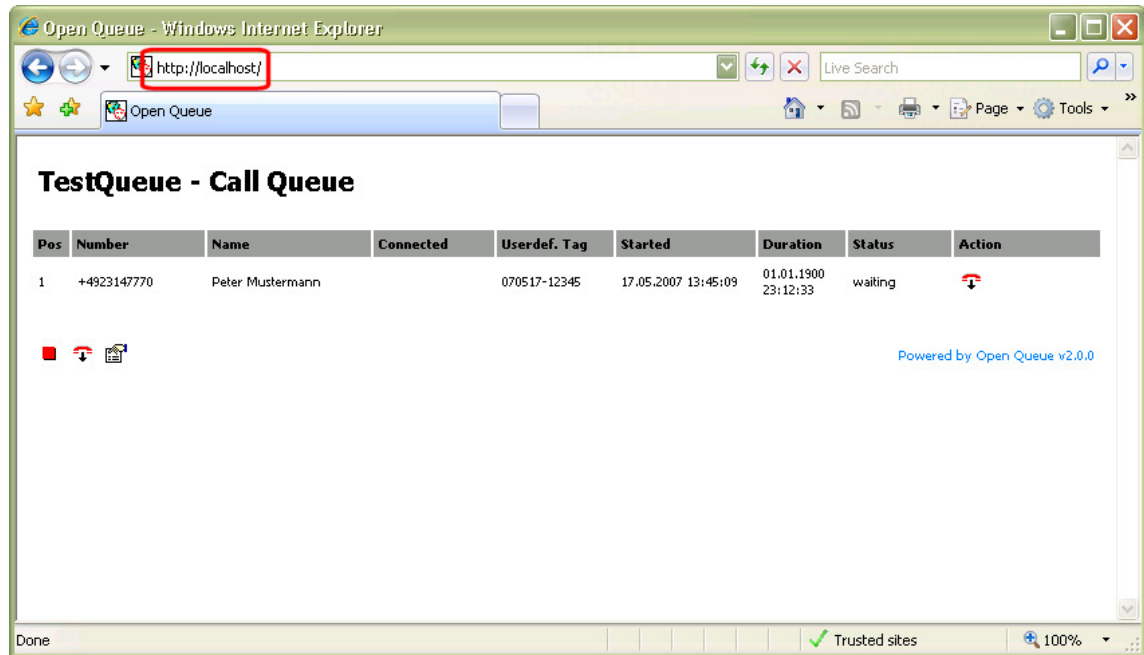
```
Const sProvider = "Provider=sqloledb;Data Source=localhost;Initial  
Catalog=OpenQueue;Integrated Security=SSPI"
```

(See the screenshot on the page behind step 9 of this instruction)

If you figure problems with “localhost” try using “ComputerName[InstanceName]”, where the instance Name is optional resp. mandatory if you are using SQL Server 2005 Express, e.g.

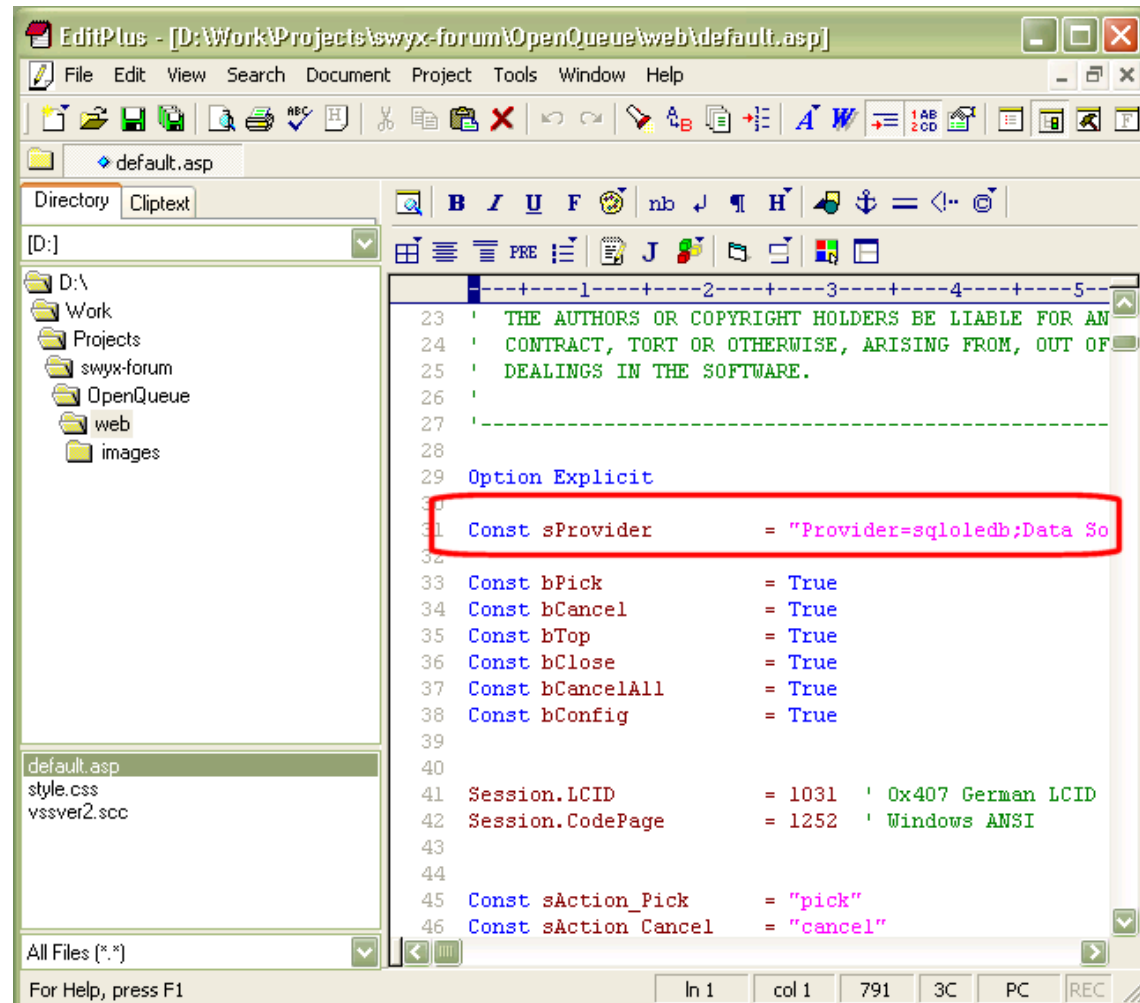
WS-WELLIGE02
NB-WELLIGE01\SQLEXPRESS

8. Open the Internet Explorer (or any other browser) and point it to localhost.



9. **Congratulations!** You have successfully configured the Internet Information Server to display the Open Queue visualization.

Please note: if you haven't installed / configured the Open Queue database as described within the previous chapter you need to update the database connect string of the web application. To do so, open the **default.asp** file in the **web** folder using a common text editor, and modify the **sProvider** constant.



```
23 ' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR AN
24 ' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
25 ' DEALINGS IN THE SOFTWARE.
26 '
27 '-----
28
29 Option Explicit
30
31 Const sProvider          = "Provider=sqloledb;Data So
32
33 Const bPick              = True
34 Const bCancel            = True
35 Const bTop               = True
36 Const bClose             = True
37 Const bCancelAll         = True
38 Const bConfig            = True
39
40
41 Session.LCID              = 1031 ' 0x407 German LCID
42 Session.CodePage         = 1252 ' Windows ANSI
43
44
45 Const sAction_Pick       = "pick"
46 Const sAction_Cancel     = "cancel"
```

If you figure problems with “localhost” try using “ComputerName[InstanceName]”, where the instance Name is optional resp. mandatory if you are using SQL Server 2005 Express, e.g.

WS-WELLIGE02

NB-WELLIGE01\SQLExpress

3.4 Updates

3.4.1 Update from v2.0 Beta 1 to RC 1

If you are already using Open Queue v2.0 Beta 1 you need to copy/replace all the **ase** and **vbs** files according to the instructions in **chapter 3.1**.

Additionally you need to update the Open Queue database from Beta 1 schema to Release Candidate 1 schema.

This needed updated script (**OpenQueue_Update_1.0.2.sql**) is located within the SQL folder of the installation package. Please refer to **chapter 3.2** - Step 14 on how to open and run an SQL script using the Microsoft SQL Server Management Studio Express.

3.4.2 Update from v2.0 RC 1 to RC 2

If you are already using Open Queue Release Candidate 1 you need to replace **two files** from the **ase** folder

- **actionOpenQueueCreateQueue.ase**
- **actionOpenQueueCreateQueue.vbs**

according to the instructions in **chapter 3.1**.

3.4.3 Update from v2.0 RC 2 to RC 3

If you are already using Open Queue Release Candidate 1 or 2 you need to replace **four files** from the **ase** folder

- **actionOpenQueueCreateQueue.ase**
- **actionOpenQueueCreateQueue.vbs**
- **actionLongestWaiting.ase**
- **actionLongestWaiting.vbs**

according to the instructions in **chapter 3.1**.

3.4.4 Update from v2.0 RC 3 to final Release

If you are already using Open Queue Release Candidate 1 or 2 you need to replace **four files** from the **ase** folder

- **actionOpenQueueCreateQueue.ase**
- **actionOpenQueueCreateQueue.vbs**
- **actionLongestWaiting.ase**
- **actionLongestWaiting.vbs**

according the instructions in **chapter 3.1**.

3.5 Longest Waiting

In order to setup the Longest Waiting feature correctly a number of more or less complex things need to be done or configured.

3.5.1 Setup Call Detail Records

As SwyxServer does not provide needed information about “Latest Disconnect Time” of a user directly, the Call Detail Records (CDRs) are used to get this information. So the CDRs must be written into the Open Queue database. Details on setting up CDRs written into a database can be taken from the Swyx Knowledgebase

Write Call Details Records into a database (kb2491)

<http://www.swyx.com/support/ssdb.html?kbid=kb2491>

The needed SQL script (**IpPbxCDR.sql**) to setup the IpPbxCDR table can be found in the SQL folder of the Open Queue installation package. Please refer to **chapter 3.2** - Step 14 on how to open and run an SQL script using the Microsoft SQL Server Management Studio Express.

3.5.2 Setup Longest Waiting Table and Trigger

Once the CDRs are written into the database (this must be the Open Queue database!) an additional table and a so called trigger must be configured. To do so, use the SQL script **LongestWaiting_1.0.0.sql** from the SQL folder of the Open Queue installation package. Please refer to **chapter 3.2** - Step 14 on how to open and run an SQL script using the Microsoft SQL Server Management Studio Express.

After having executed this script the created trigger will automatically take newly written CDRs from the SwyxServer, extract the user and disconnect time from it and stores these information within the LongestWaiting table.

In order to get this working correctly two parameters need to be configured within the Parameters table:

ExtensionRange	the number of digits of SwyxWare’s extension range, default: 3
PublicLineAccess	number to be dialed for public line access, default: default: 0

Once everything is configured correctly you can check the proper functionality by having a look into the LongestWaiting table to check if records for current timestamps will be written.

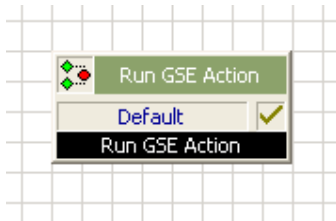
3.5.3 Setup Longest Waiting Scripts

This in fact has already happened by following the setup instructions in **chapter 3.1** or the update instructions in **chapter 3.4**.

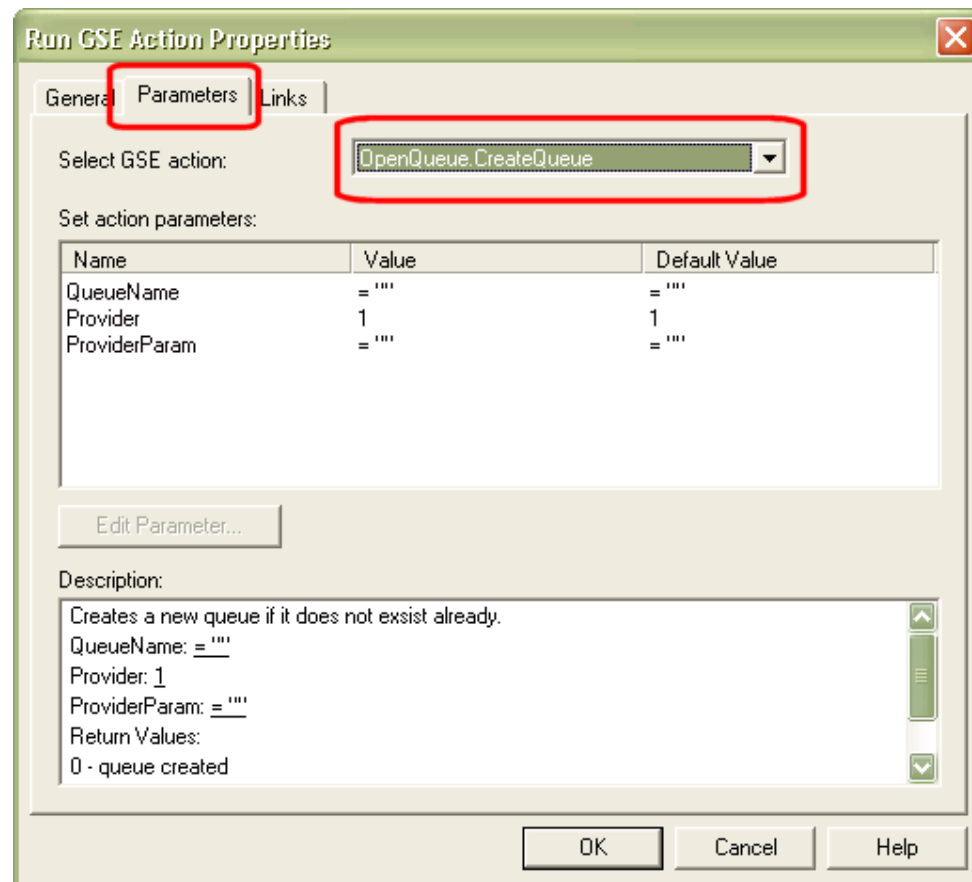
4 Usage

4.1 How to use Open Queue actions

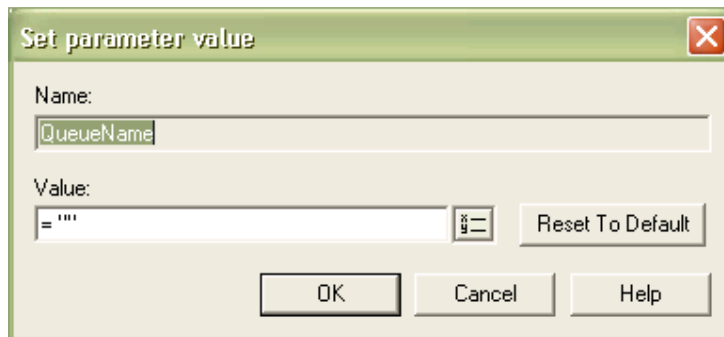
Unlike common GSE functions there are no ready made blocks for Open Queue available. Open Queue comes as a set of GSE actions that can be used in any GSE script. To call a GSE action you have to use a **Run GSE Action** block.



Open the **Properties** of this block and switch to the **Parameters** page. Select the wanted Open Queue action.

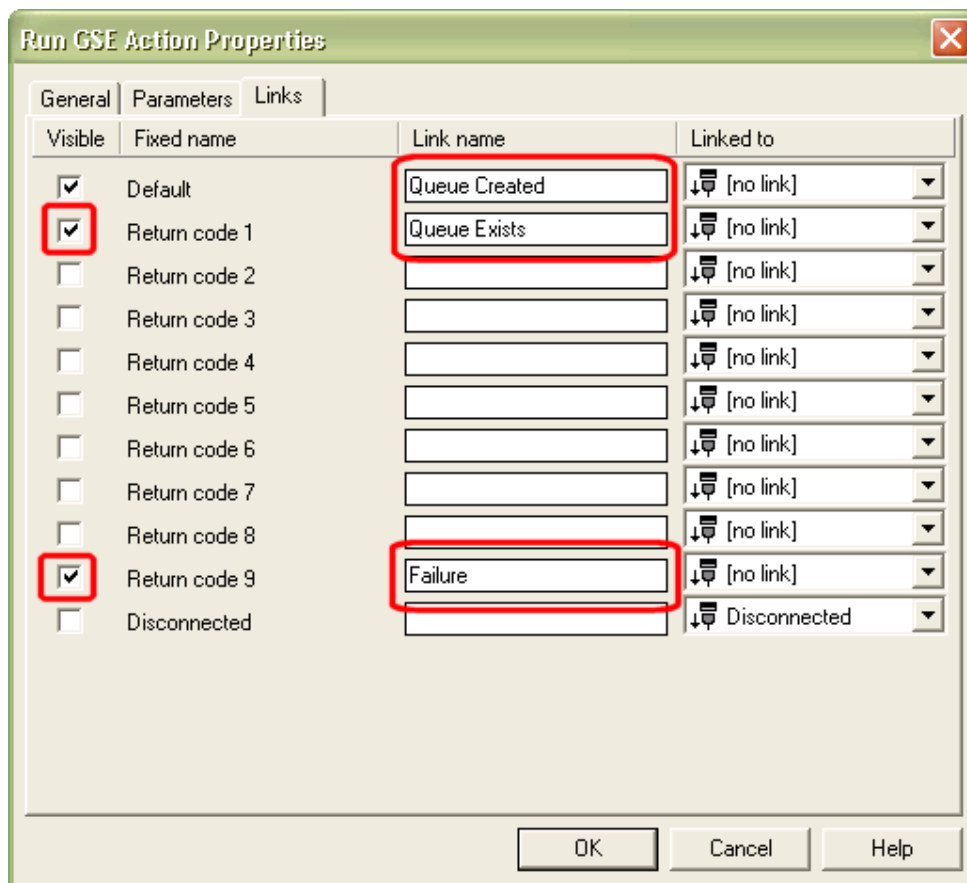


By double clicking on each **parameter** you will be able to modify it.



The 'Set parameter value' dialog box has a title bar with a close button. It contains two input fields: 'Name:' with the text 'QueueName' and 'Value:' with the text '='. To the right of the 'Value:' field is a 'Reset To Default' button. At the bottom are three buttons: 'OK', 'Cancel', and 'Help'.

Finally make sure all **Return Values** of the selected action will be displayed as exits. To do so switch to the **Links** page, **enable** the needed exits and set proper **names** for the exits:



The 'Run GSE Action Properties' dialog box has three tabs: 'General', 'Parameters', and 'Links'. The 'Links' tab is selected. It contains a table with four columns: 'Visible', 'Fixed name', 'Link name', and 'Linked to'. The 'Visible' column has checkboxes for 'Default', 'Return code 1', 'Return code 2', 'Return code 3', 'Return code 4', 'Return code 5', 'Return code 6', 'Return code 7', 'Return code 8', 'Return code 9', and 'Disconnected'. The 'Fixed name' column contains the corresponding names. The 'Link name' column has text boxes for each row, with 'Queue Created', 'Queue Exists', and 'Failure' entered. The 'Linked to' column has dropdown menus, all set to '[no link]' except for 'Disconnected' which is set to 'Disconnected'. Red boxes highlight the checkboxes for 'Return code 1' and 'Return code 9', and the text boxes for 'Queue Exists' and 'Failure'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Visible	Fixed name	Link name	Linked to
<input checked="" type="checkbox"/>	Default	Queue Created	[no link]
<input checked="" type="checkbox"/>	Return code 1	Queue Exists	[no link]
<input type="checkbox"/>	Return code 2		[no link]
<input type="checkbox"/>	Return code 3		[no link]
<input type="checkbox"/>	Return code 4		[no link]
<input type="checkbox"/>	Return code 5		[no link]
<input type="checkbox"/>	Return code 6		[no link]
<input type="checkbox"/>	Return code 7		[no link]
<input type="checkbox"/>	Return code 8		[no link]
<input checked="" type="checkbox"/>	Return code 9	Failure	[no link]
<input type="checkbox"/>	Disconnected		Disconnected

The **chapter 5** for a complete list of Open Queue GSE actions, their functionality, their parameters and return values.

4.2 Simple Call Routing Script

The following is a simple queue script defining a queue named “Support”. You’ll find it within the rse folder of the Open Queue package. To load it, open a new GSE script and import the Simple.rse file via the File menu.

The queue connects calls to the “Support” user group and uses the newly introduced music on hold “cm_Dream Traveller.wav”. The file can be found in the

```
C:\Documents and Settings\All Users\
Application Data\Swyx\Share\Data\System\HoldMusic\
```

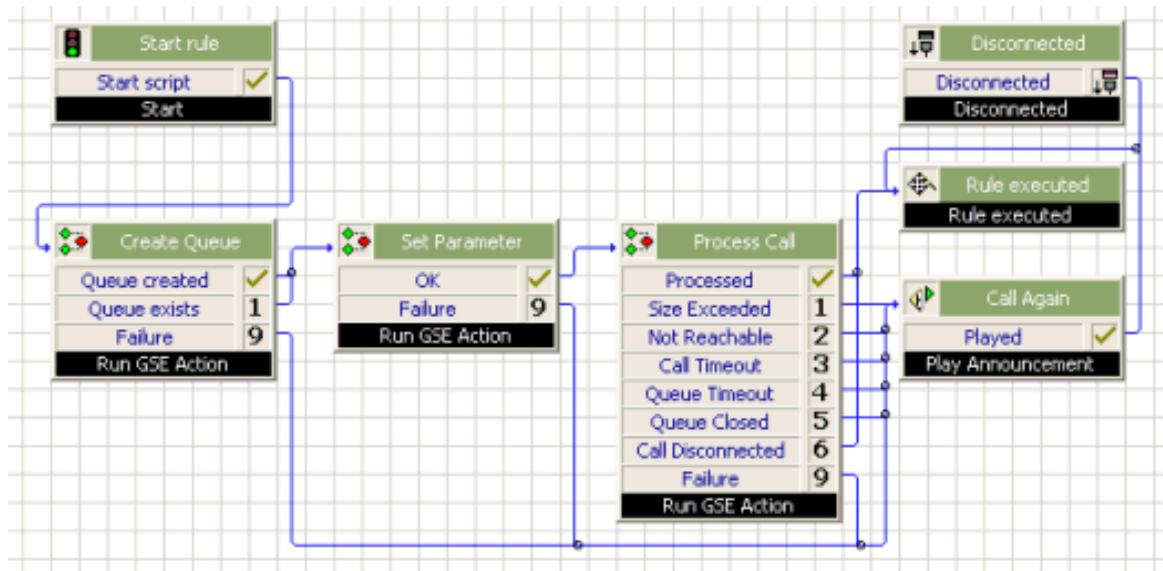
of any SwyxWare v6.02 (or newer).

IMPORTANT: You definitely have to alter the ProviderParameter (DB Connect String) of the “Create Queue” action block. You have to set the name of the machine the Open Queue database is running on instead of localhost

```
Provider=sqloledb;Data Source=localhost;Initial Cata-
log=OpenQueue;Integrated Security=SSPI;
```

If you figure problems with “localhost” try using “ComputerName[InstanceName]”, where the instance Name is optional resp. mandatory if you are using SQL Server 2005 Express, e.g.

```
WS-WELLIGE02
NB-WELLIGE01\SQLEXPRESS
```



You should alter the queue parameters according your needs. You can use any number of different queues within a single ECR script. There is also no limitation of queue on the entire Swyx Server.

4.3 Queue Visualization







If you have followed the instructions in chapter 3.3 you will be able to use the Open Queue visualization. This is a small web application displaying all calls in one particular or all call queues. Within the call list you will be able to

- Pick a call from the list.
- Cancel (disconnect) a call from the list.
- Move a call to the top position of the queue.
- Cancel (disconnect) all calls from a queue.
- Close a queue to prevent new calls to be added to it.
- List all configured queue parameters.

Per call you will see

- its position
- the caller number (if available)
- the caller name (as resolved by Swyx Server)
- the name of the user who has been connected to the call
- user defined data (if has been set when adding the call to the queue)
- timestamp when the call was added to the queue
- duration of the call within the queue.

Support - Call Queue

Pos	Number	Name	Connected	Userdef. Tag	Started	Duration	Status	Action
1	+4923147770	Peter Mustermann	Tom Wellige	070517-12345	17.05.2007 13:45:09	00:17:09	connected	
2	+49405566881			070517-54321	17.05.2007 13:59:09	00:03:09	waiting	  
3	+49231123456	Erika Mustermann		070517-11223	17.05.2007 14:00:43	00:01:35	waiting	  




Powered by Open Queue v2.0.0


Max Size:	10
Timeout:	300
Connect Timeout:	10
Connect Alert Sound:	<< use system alert sound >>
Destinations:	100
Use Longest Waiting:	False
Music On Hold:	Dream Traveller.wav
Wait Announcement:	<< no wait announcement >>
Pre-Position Announcement:	<< no pre-position announcement >>
Post-Position Announcement:	<< no post-position announcement >>
Wait/Position Timeout:	15
Disconnect Announcement:	<< no disconnect announcement >>

4.3.1 Functions


4.3.1.1 Pick call from queue

If the page was called including the “target” URL parameter, defining the “Pick” target, the Pick Call icon  will be displayed. Additionally the call must be in wait state. By clicking on this icon the call will be diverted to the target number.


4.3.1.2 Cancel call from queue

If the call is in wait state the “Cancel Call” icon  will be displayed. By clicking on it the call will be disconnected. If a disconnect announcement has been configured this will be played previously.


4.3.1.3 Move to top position in queue

If the call is not on the top position of the queue and in wait state it can be moved to the top position of the queue by clicking on the “Move to Top” icon .


4.3.1.4 Close queue

By clicking on the Close Queue icon  you close the queue. A closed queue will not take any new calls.

4.3.1.5 Open queue

By clicking on the Open Queue icon  you open a previously closed queue. The queue will now accept new calls again.

4.3.1.6 Cancel all calls in queue

By clicking on the Cancel all Calls icon  all current calls within the queue will be disconnected. If a disconnect announcement has been configured this will be played previously.

4.3.1.7 Show / Hide queue configuration

By clicking on the Show / Hide Configuration icon  the complete queue configuration (parameters) will be displayed, or not.

4.3.2 URL Parameter

You can call the default.asp page without any parameters. Doing this you get all configured queue displayed with all function icon except the “Pick Call” icon, as there is no pick target set.

You can set additional URL parameters to change this.

4.3.2.1 target

The target parameter defines the pick target. This is either a number or user name.

```
http://localhost/default.asp?target=123
```

4.3.2.2 queueName

If you want to display a particular queue only (and not all queues) you can use the queueName parameter. This is the name of the queue to be displayed only.

```
http://localhost/default.asp?queueName=Support
```

Of course you can use a combination of these parameters as well !

```
http://localhost/default.asp?target=123&queueName=Support
```

4.3.3 Functionality Configuration

Beside the URL parameters you have a set of other configuration possibilities. These are hard coded within the default.asp file. For each of the functions listed in chapter 4.2.1 there is a Boolean constant to enable or disable a feature.

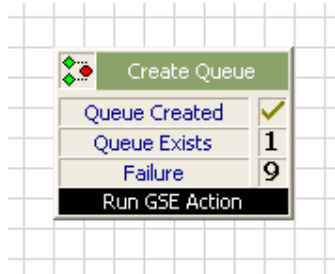
```
Const bPick           = True
Const bCancel         = True
Const bTop            = True
Const bClose          = True
Const bCancelAll      = True
Const bConfig         = True
```

To disable a certain function just set the value from True to False.

With this it is possible to create one asp file being displayed for common call center agents with Pick Call and Move to Top icons. Another asp file being displayed for supervisors would have all functions enabled.

5 GSE Actions

5.1 OpenQueue.CreateQueue



Creates a new queue if it does not exist already.

This action is mandatory for each script making use of Open Queue and must be called before all other Open Queue actions.

When using the Generic Database Provider this action creates a new record within the OpenQueue_Config table.

5.1.1 Parameter

- **QueueName**
Name of the queue. This name is used in all actions to identify the queue to be used. The name can be any alpha-numeric value, up to 50 characters.
- **Provider**
Selects the data provider being used by Open Queue. This is a numeric value. Valid values:

1 – Generic Database Provider
- **ProviderParam**
Passes additional provider parameters. In case of the Generic Database Provider this is the connect string to the database. Examples:

Example 1: SQL Server/MSDE on same machine as SwyxServer, using Windows Security for database login (IpPbxLogin has access to OpenQueue database)

```
Provider=sqloledb;Data Source=localhost;  
Initial Catalog=OpenQueue;Integrated Security=SSPI
```

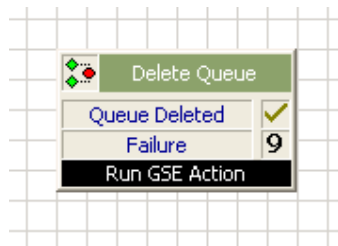
Example 2: SQL Server/MSDE on different machine as SwyxServer, using dedicated SQL login:

```
Provider=sqloledb;Data Source=myserver;  
Initial Catalog=OpenQueue;User Id=openqueue;Password=openqueue
```

5.1.2 Return Values

- **0** – Queue successfully created.
- **1** – Queue already exists, no action was taken.
- **9** – Failure, most likely a provider (database) problem, see SwyxServer trace (SrvScript, INFO3 i.e. 6) for details.

5.2 OpenQueue.DeleteQueue



Deletes the given queue. This action is used to delete a queue from the provider's storage. Unless you don't want to wipe a test or unused call queue from the provider's storage you don't need to call this action.

When using the Generic Database Provider this action deletes all calls from the QueueConfig_Calls table and the queue config record from the OpenQueue_Config table.

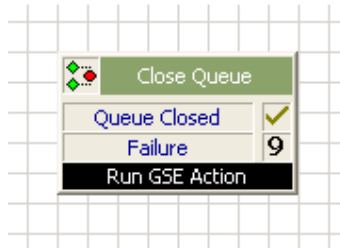
5.2.1 Parameter

- **QueueName**
Name of the queue to be deleted.

5.2.2 Return Values

- **0** – Queue successfully deleted.
- **9** – Failure, most likely a provider (database) problem, see SwyxServer trace (SrvScript, INFO3 i.e. 6) for details.

5.3 OpenQueue.CloseQueue



Closes the given queue. This action is called at the final end of a ECR script, but **only** if the used provider needs to release some used resources.

The Generic Database Provider needs not to be closed at the end of an ECR script, as it will be initialized with a parameter telling him to open and close the database connection for each single provider function call.

If you want to force the Generic Database Provider to leave the database connection open instead of opening and closing it for each function call you need to modify the COpenQueue.CreateQueue function which is defined in the start block of the Create Queue action. Search for

```
nReturn = m_cProvider.CreateQueue ( nCurrentQueueID, sQueueName,  
sProviderParam, False )
```

and set the latest parameter from `False` to `True`.

5.3.1 Return Values

- **0** – Queue successfully closed.
- **9** – Failure, most likely a provider (database) problem, see SwyxServer trace (SrvScript, INFO3 i.e. 6) for details.

5.4 OpenQueue.SetParameter



This action sets the given queue parameters. Depending on the used provider this action must be called in each script or just once in any script and the provider keeps the settings permanently.

When using the Generic Database Provider you need to call this action once right after the first Create Queue call. Afterwards you can skip it and call it if you want to change the parameters only.

The Generic Database Provider stores the queue parameters within the OpenQueue_Config table.

5.4.1 Parameter

- **QueueName**
Name of queue the parameters should be set for.
- **MaxSize**
Maximum number of calls handled by the queue. This is a numeric value, the default value is 10. When trying to process the MaxSize+1 call with the **ProcessCallByQueue** action you will end up immediately in the “Size Exceeded” (1) exit.
- **TimeOut**
Number of seconds a call should remain within the queue. This is a numeric value, the default value is 300 (i.e. 5 minutes). When the timeout is reached you will end up in the **ProcessCallByQueue** “Queue Timeout” (4) exit.
- **ConnectTimeOut**
Number of seconds to try to connect a call to the destination once the destination is detected to be available. During this time the destination will alert. This is a numeric value, the default value is 10. When the timeout is reached without the destination has accepted the call, the call will placed back into the queue (to its original position).
- **ConnectAlertSound**
Name of the wav file to be played to the caller while connecting a call to the destination. This is a string value with a maximum length of 100 characters. The default value is an empty string , meaning the systems alert sound will be played. The wav file must be placed in the script user’s voicemail folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\User\nnn\PhoneClient\Voicemail
```

where `nnn` is the name of the script user.

or the global announcement folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\Data\PhoneClient\VoiceMail\
```

- **Destination**

Number of name of the queue's destination, meaning the user or group taking all calls to the group. This is a string value, the maximum length is 50 characters. If the destination is a group of users (that's most likely the case) you'll configure the hunt group settings (parallel, sequential, rotary or random) within the user group properties in the SwyxWare Administration. If you want to use a "Longest Waiting" hunt group read **chapter 3.4** for special setup instructions and use additionally the UseLongestWaiting parameter explained later on.

- **Closed**

It is possible to close the queue using this parameter. This is a numeric value, the default value is 0. The possible values are:

0 – Queue opened, new calls will be processed by the **ProcessCallByQueue** action.

1 – Queue is closed, new calls will not processed by the **ProcessCallByQueue** action but instead you will end up in it's "Queue Closed" (5) exit.

This value is usually been set from outside the call routing, e.g. from the queue visualization, but it is also possible to close the queue from within the call routing.

- **UseLongestWaiting**

If this parameter is set the **ProcessCallByQueue** action will use the special "Longest Waiting" feature instead of the standard SwyxWare connect mechanism to connect a call to the destination. If you want to use a "Longest Waiting" hunt group read **chapter 3.5** for special setup instructions. This is a numeric value, the default value is 0. The possible values are:

0 – Use the standard SwyxWare connect mechanism with the hunt group configuration within the user group properties.

1 – Use the Open Queue built in "Longest Waiting" connect mechanism.

- **WaitForDisconnect**

After the **ProcessCallByQueue** action has connected a call to the destination either wait for disconnect of the call or return immediately through it's "Processed" (0) exit. If this feature is not enabled you need to use the Open Queue's **Wait for Disconnect** block within your call routing script and call afterwards the **RemoveCallFromQueue** action. Otherwise the call won't be deleted from the queue management, even though it's already disconnected. This is a numeric value, the default value is 1. The possible values are:

0 – Do not wait from disconnect within the ProcessCallByQueue action. Do not forget to call the RemoveCallFromQueue by yourself!

1 – Wait for disconnect within the ProcessCallByQueue action. In this case you don't need to worry about removing the call from the queue.

- **MusicOnHold**

This is the name of the wav file to be played while waiting within the queue. The file will automatically be repeated. This is a string value, the maximum length is 100 characters. The wav file must be placed in the script user's voicemail folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\User\nnn\PhoneClient\Voicemail
```

where `nnn` is the name of the script user.
or the global announcement folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\Data\PhoneClient\VoiceMail\
```

- **WaitAnnouncement**

This is the name of the wav file playing an announcement like "We are sorry, all agents are busy, please wait." This announcement will be played once before the music on hold is started and will be repeated regularly during music on hold. See the **WaitPosTimeout** parameter for the timeout the announcement is repeated. During this announcement is played the music on hold is faded down. When leaving this parameter empty no announcement will be played. This is a string value, the maximum length is 100 characters. The wav file must be placed in the script user's voicemail folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\User\nnn\PhoneClient\Voicemail
```

where `nnn` is the name of the script user.
or the global announcement folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\Data\PhoneClient\VoiceMail\
```

- **PrePosAnnouncement**

The queue is able to announce the current caller's position within the queue. If this or the next parameter, or both are configured this feature is enabled. Without setting this or / and the next parameter this feature is disabled. This parameter defines an announcement that is played right **before** the system announces the position. During the complete position announcement the music on hold is faded down. The position announcement will be play directly after the wait announcement (if configured). See the **WaitPosTimeout** parameter for the timeout the position announcement is repeated. This is a string value, the maximum length is 100 characters. The wav file must be placed in the script user's voicemail folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\User\nnn\PhoneClient\Voicemail
```

where `nnn` is the name of the script user.
or the global announcement folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\Data\PhoneClient\VoiceMail\
```

- **PostPosAnnouncement**

The queue is able to announce the current caller's position within the queue. If this or the next parameter, or both are configured this feature is enabled. Without setting this or / and the next parameter this feature is disabled. This parameter defines an announcement that is played right **after** the system announces the position. During the complete position announcement the music on hold is faded down. The position announcement will be play directly after the wait announcement (if configured). See the **WaitPosTimeout** parameter for the timeout the position announcement is repeated. This is a string value, the maximum length is 100 characters. The wav file must be placed in the script user's voicemail folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\User\nnn\PhoneClient\Voicemail
```

where `nnn` is the name of the script user.
or the global announcement folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\Data\PhoneClient\VoiceMail\
```

- **WaitPosTimeout**

This is the number of seconds the wait and / or position announcement will be repeated. This is a numeric value, the default value is 15. If neither the wait announcement nor the position announcement is configured this parameter doesn't mean anything.

- **DisconnectAnnouncement**

It is possible to disconnect a call within the queue from outside, e.g. the queue visualization. This parameter defines the announcement that is played right after initiating canceling of the call but right before doing actually. This is a string value, the maximum length is 100 characters. If this parameter is left empty no announcement will be played. The wav file must be placed in the script user's voicemail folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\User\nnn\PhoneClient\Voicemail
```

where `nnn` is the name of the script user.
or the global announcement folder:

```
C:\Documents and Settings\All Users\Application Data\
Swyx\Share\Data\PhoneClient\VoiceMail\
```

5.4.2 Return Values

- **0** – All parameters successfully set.
- **9** – Failure, most likely a provider (database) problem, see SwyxServer trace (SrvScript, INFO3 i.e. 6) for details.

5.5 OpenQueue.GetParameter



This action returns certain or all parameters for given queue. You can call this action to retrieve the configured parameters at any time.

The action writes the queue parameters into given variables. If you do not want to obtain certain parameters just leave the default value of the action's parameters instead of passing a variable name. So it is possible to just get the current size of a queue without worrying about all the other parameters.

5.5.1 Parameter

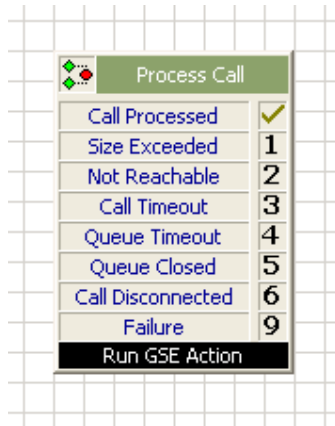
- **QueueName**
Name of queue the parameters should be set for.
- **CurrentSize**
Number of calls currently within the queue. This is a numeric value.
- **MaxSize**
See **chapter 5.4.1** for details.
- **TimeOut**
See **chapter 5.4.1** for details.
- **ConnectTimeOut**
See **chapter 5.4.1** for details.
- **ConnectAlertSound**
See **chapter 5.4.1** for details.
- **Destination**
See **chapter 5.4.1** for details.
- **Closed**
See **chapter 5.4.1** for details.
- **UseLongestWaiting**
See **chapter 5.4.1** for details.
- **WaitForDisconnect**
See **chapter 5.4.1** for details.

- **MusicOnHold**
See **chapter 5.4.1** for details.
- **WaitAnnouncement**
See **chapter 5.4.1** for details.
- **PrePosAnnouncement**
See **chapter 5.4.1** for details.
- **PostPosAnnouncement**
See **chapter 5.4.1** for details.
- **WaitPosTimeout**
See **chapter 5.4.1** for details.
- **DisconnectAnnouncement**
See **chapter 5.4.1** for details.

5.5.2 Return Values

- **0** – All parameters successfully returned.
- **9** – Failure, most likely a provider (database) problem, see SwyxServer trace (SrvScript, INFO3 i.e. 6) for details.

5.6 OpenQueue.ProcessCallByQueue



This action processes current call by queue. Make sure you have configured all queue parameters before calling this action. The following is what this action does in detail:

- Check if destination is reachable directly. If yes connect the call using the configured connect mechanism (standard SwyxWare or “Longest Waiting”).
- Check if the configured maximum call queue size (queue parameter) is reached and if so exit the action via it’s “Size Exceeded” (1) exit.
- Check if call is closed and if so exit the action via it’s “Queue Closed” (5) exit.
- Add the current call to the given position into the queue (tail or bottom, this action’s parameter).
- Play wait announcement (if configured, queue parameter)
- Start the Music on hold.
- Within a **one second** cycle do
 - If current call is on top position of the queue and the destination is available (free, not line keys locked) connect the call using the configured connect mechanism (standard SwyxWare or “Longest Waiting”). If the call is not connected within the configured connect timeout (queue parameter) set the call status back to waiting.
 - If the call has to be diverted (pick call, configured from outside the queue, e.g. from the queue visualization) connect the call to the divert target using the configured connect mechanism (standard SwyxWare or “Longest Waiting”). If the call is not connected within the configured connect timeout (queue parameter) set the call status back to waiting.
 - If the call has to be disconnected (cancel call, configured from outside the queue, e.g. from the queue visualization) play the disconnect announcement (if configured, queue parameter) and disconnect the call.

- If WaitPosTimeout (queue parameter) is reached
 - play wait announcement (if configured, queue parameter).
 - play position announcements (if configured, queue parameter)
- If queue timeout (queue parameter) is reached stop the music on hold and exit the action via it's "Queue Timeout" (4) exit.
- If call timeout (this action's parameter) is reached stop the music on hold and exit the action via it's "Call Timeout" (3) exit.

If the call was connected, regardless if the call was on top position of the queue or was picked from the queue, the action will exit through it's "Processed" (0) exit. If the wait for disconnect parameter (queue parameter) was set (default), this will happen only after the call has been disconnected.

If the destination is not reachable (invalid destination number/name, no user logged in) the action will exit through it's "Not Reachable" (2) exit.

5.6.1 Parameter

- **QueueName**
Name of the queue to process the current call with.
- **CallID**
This is an out parameter. It returns the provider's internal id for this call. Do not mistake this value with the SwyxWare Call ID. If you have disabled "Wait for disconnect" via a queue parameter you need this id to remove it from the queue manually using the **RemoveCallFromQueue** action. Use it as following:
 - create a new variable, e.g. QueueCallID, with a **Set Variable** block
 - pass this variable name as CallID parameter of the **ProcessCallByQueue** action
 - use the **Wait for Disconnect** block to wait for the actual disconnection of the call
 - call the **RemoveCallFromQueue** action passing the QueueCallID parameter to remove the call from the queue.

Again: this is only needed to be done if you do not use the default queue handling with activated wait for disconnect within the **ProcessCallByQueue** action.

- **Tag**
You can store any kind of string data (up to 100 characters) for each call within the queue. This might be something like a support ticket id the caller has entered previously via dtmf. The queue visualization can be used to display this user defined data (tag). The queue itself doesn't consider this data at all. This is a string value, the maximum length is 100 characters.

- **Timeout**

Beside the queue timeout you can define also a call timeout for a single call in the queue. If this timeout is reached, i.e. the caller has waited for this amount of seconds within the queue, the action will exit though it's "Call Timeout" (3) exit. This is might be useful for VIP calls you want to handle in your standard support queue. Instead of passing the queue timeout you pass a shorter timeout to make sure a VIP will be handled faster if there isn't an agent available. This is a numeric value. It defines the number of seconds for the timeout.

- **Position**

When adding a call to the queue you decide if the call should be added at the end of the queue (default) or it's top. This is a numeric value. The following values are possible:

0 – Add call to tail of queue (default).

1 – Add call to the top of the queue.

5.6.2 Return Values

- **0** – Call processed by queue. If "WaitForDisconnect" was enabled (default) this exit will be reached, once the call was also disconnected. Otherwise this exit will be reached immediately after the connect to the destination.
- **1** – The maximum size of the queue has been reached. The current call hasn't been added to the queue.
- **2** – Destination is not reachable. This exit will be reached if the destination is an invalid number/name or no user is logged in within the destination group.
- **3** – Call timeout has been reached.
- **4** – Queue timeout has been reached.
- **5** – The queue is closed. The current call hasn't been added to the queue.
- **6** – The call has been disconnected and removed from the queue.
- **9** – Failure, most likely a provider (database) problem, see SwyxServer trace (SrvScript, INFO3 i.e. 6) for details.

5.7 OpenQueue.RemoveCallFromQueue



This action removes the current call from queue. You need to call the action only if you have disabled the Wait for Disconnect parameter of the queue. Per default the Wait for Disconnect parameter is enabled, so you don't need to call this action.

5.7.1 Parameter

- **QueueCallID**
The queue's call id as being returned by the **ProcessCallByQueue** action.

5.7.2 Return Values

- **0** – Call successfully removed from queue.
- **9** – Failure, most likely a provider (database) problem, see SwyxServer trace (SrvScript, INFO3 i.e. 6) for details.

5.8 OpenQueue.WaitForDisconnect



When using the `ProcessCallByQueue` action with the parameter `WaitForDisconnect = 0` (Default: 1) you have to wait for the final disconnect of the call and afterwards remove the call from the queue by using the `RemoveCallFromQueue`. You **must not** use the GSE's build-in block `WaitForDisconnect` but instead use this action. This is necessary for the build-in "keep-alive" mechanism of Open Queue. If you use the GSE's `WaitForDisconnect` block, the current call would be disconnected after 30 seconds !

5.8.1 Parameter

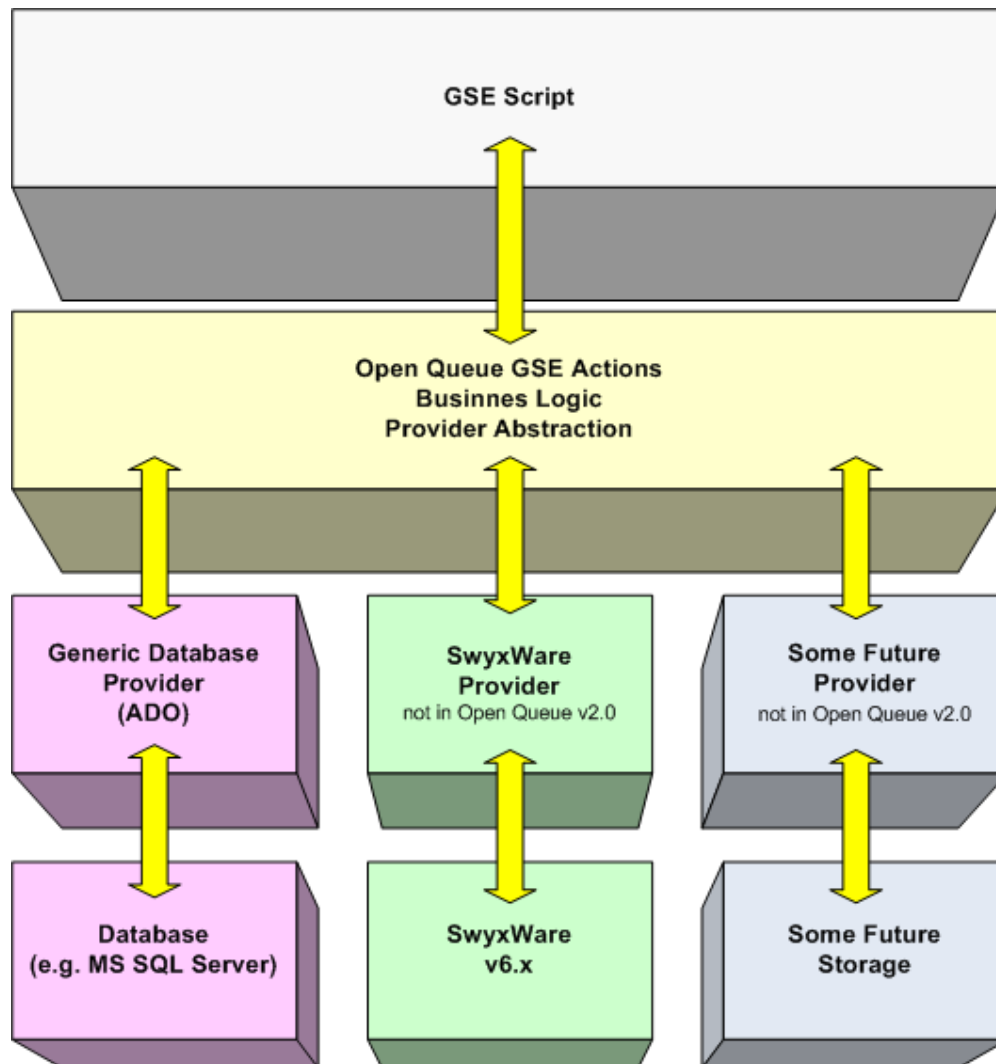
- **QueueCallID**
The queue's call id as being returned by the **ProcessCallByQueue** action.

5.8.2 Return Values

- **0** – Call was disconnected.
- **9** – Failure, most likely a provider (database) problem, see SwyxServer trace (SrvScript, INFO3 i.e. 6) for details.

6 Reference

6.1 Programming model



6.2 Business Logic

6.2.1 COpenQueue Class

6.2.1.1 Constants

Providers

cOpenQueue_UseGenericProvider	1
cOpenQueue_UseSwyxQueueManager	2

Call States

cOpenQueue_CallState_Waiting	1
cOpenQueue_CallState_Alerting	2
cOpenQueue_CallState_Connected	3
cOpenQueue_CallState_Diverted	4
cOpenQueue_CallState_Disconnected	5
cOpenQueue_CallState_Disconnect	6

Return Codes

cOpenQueue_ErrorNone	0
cOpenQueue_Processed	0
cOpenQueue_QueueExists	1
cOpenQueue_SizeExceeded	1
cOpenQueue_NotReachable	2
cOpenQueue_Timeout	3
cOpenQueue_QueueTimeout	4
cOpenQueue_Closed	5
cOpenQueue_Disconnected	6
cOpenQueue_Error	9
cOpenQueue_ErrorProvider	9

6.2.1.2 Public Properties

Version
Provider
ProviderCallID

6.2.1.3 Public Methods

OpenQueue_CreateQueue
OpenQueue_DeleteQueue
OpenQueue_CloseQueue
OpenQueue_SetParameter
OpenQueue_GetParameter
OpenQueue_ProcessCallByQueue
OpenQueue_RemoveCallFromQueue
OpenQueue_WaitForDisconnect

6.3 Provider

A provider is defined in it's own VBScript file. This file is included via the SwyxWare's own include statement into the start block of the **Create Queue** action. As the provider uses it's own file, it must be digitally signed in order to get loaded from the SwyxWare server.

If you modify this file or want to write your own provider class you need the SignScript tool to updated/set the signature. You'll find this tool in the Swyx Partner Net following this URL:

<http://www.swyx.com/partnet/ssdb.html?kbid=kb2635>

6.3.1 COpenQueue_GenericDBProvider Class

This is the **Generic Database Provider**.

6.3.1.1 Public Properties

Name
Version

6.3.1.2 Public Methods

CreateQueue
DeleteQueue
CloseQueue
SetParameter
GetParameter
AddCallToQueue
RemoveCallFromQueue
GetPosition
GetCallDiverted
SetCallState
GetCallState
SetConnectedInfo

6.3.2 Database

6.3.2.1 OpenQueue_Config Table

Field	Data type	Description
ID	Int	unique id
QueueName	Varchar 50	queue name, given by admin
MaxSize	Int *	maximum number of calls in queue
TimeOut	Int *	queue timeout
ConnectTimeOut	Int *	connect timeout
ConnectAlertSound	Varchar 100 *	connect alert sound, use system alert sound if empty
Destination	Varchar 50 *	destination number, name of user or group
Closed	Bit *	decline new calls in queue
UseLongestWaiting	Bit *	use "LongestWaiting" action instead of ConnectTo
WaitForDisconnect	Bit *	after successful connect to agent wait for disconnect
MusicOnHold	Varchar 100 *	name of wav file for music on hold
WaitAnnouncement	Varchar 100 *	name of wav file for "please wait" announcement
PrePosAnnouncement	Varchar 100 *	name of wav file for pre position announcement
PostPosAnnouncement	Varchar 100 *	name of wav file for post position announcement
WaitPosTimeout	Int *	timeout in seconds to play wait and/or position announcements
DisconnectAnnouncement	Varchar 100 *	name of wav file to be played on forced disconnect

6.3.3 OpenQueue_Calls Table

Field	Data type	Description
ID	Int	unique id
QueueID	Int	call for queue id
CallID	Int *	SwyxWare's CallID for current call
CallerName	Varchar 100 *	SwyxWare's Caller Name
CallerNumber	Varchar 100 *	SwyxWare's Caller Number
ConnectedNumber	Varchar 100 *	call connected with number
ConnectedName	Varchar 100 *	call connected with name
Tag	Varchar 100 *	Userdefined Tag
Timeout	Int *	Timeout for this call
Created	DateTime *	TimeStamp when call came in
OrderCreated	DateTime *	Timestamp to sort calls in queue after, might be manipulated to alter ordering
Status	Int *	current call status like
DivertTo	Varchar 50 *	target the call should be diverted to (pick from queue)
Watchdog	DateTime *	Timestamp being updated every one or five seconds as long as the call is still active. Is used to identify "hanging" calls within the queue.

6.3.4 OpenQueue_Status Table

Field	Data type	Description
ID	Int	Unique ID
Name	Varchar 20	Name of status (see call state constants)

When using the **Longest Waiting** feature the following tables are present as well (see **chapter 3.5**):

6.3.5 IpPbxCDR

Field	Data type	Description
CallId	Int	
OriginationNumber	Nvarchar 50 *	
OriginationName	Nvarchar 80 *	
CalledNumber	Nvarchar 50 *	
CalledName	Nvarchar 80 *	
DestinationNumber	Nvarchar 50 *	
DestinationName	Nvarchar 80 *	
StartTime	DateTime *	
ScriptConnectTime	DateTime *	
DeliveredTime	DateTime *	
ConnectTime	DateTime *	
EndTime	DateTime *	
Currency	Nvarchar 5 *	
Costs	Nvarchar 10 *	
State	Nvarchar 15 *	
PublicAccessPrefix	Char 1 *	
LCRProvider	Nvarchar 10 *	
ProjectNumber	Nvarchar 50 *	
AOC	Bit *	
OriginationDevice	Nvarchar 50 *	
DestinationDevice	Nvarchar 50 *	
TransferredByNumber	Nvarchar 50 *	
TransferredByName	Nvarchar 80 *	
TransferredCallId1	Int *	
TransferredCallId2	Int *	
TransferredToCallId	Int *	
TransferTime	DateTime *	
DisconnectReason	Nvarchar 50 *	

This table also has a trigger **NEW_CDR**. This trigger takes the extension and disconnect time from a cdr record and copies it into the LongestWaiting table.

See the Swyx Knowledgebase for a detailed description of all fields of this table

Call Detail Record (CDR) Format v4.10 (kb2413)
<http://www.swyx.com/support/ssdb.html?kbid=kb2413>

6.3.6 LongestWaiting Table

Field	Data type	Description
agent_id	Nvarchar 50	Internal Extension
latest_disconnect	DateTime	Latest Disconnect Time of above extension.

This table will be automatically populated by the **NEW_CDR** Trigger of the IpPbxCDR table.