

Swyx Trace Tool Server Installation

Overview

The upload functionality of Swyx Trace Tool uses Microsoft Background Intelligent Transfer Service (BITS) to perform uploads. Details about BITS can be found here:

<http://www.microsoft.com/windowsserver2003/techinfo/overview/bits.mspx>

Installation information about the BITS Server Extensions can be found here:

<http://technet2.microsoft.com/windowsserver/en/library/635498a3-709e-4592-aa1c-21c4182f30731033.mspx?mfr=true>

Configuration information for BITS uploads can be found here:

[http://msdn2.microsoft.com/en-us/library/aa363157\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/aa363157(VS.85).aspx)

On the client side Swyx Trace Tool uses the BITS API to instruct Windows BITS Service to perform the file upload. To receive uploads the server needs to have IIS running and BITS Server Extensions installed and configured. This document describes how to setup BITS on a Windows Server 2003 with IIS 6.0.

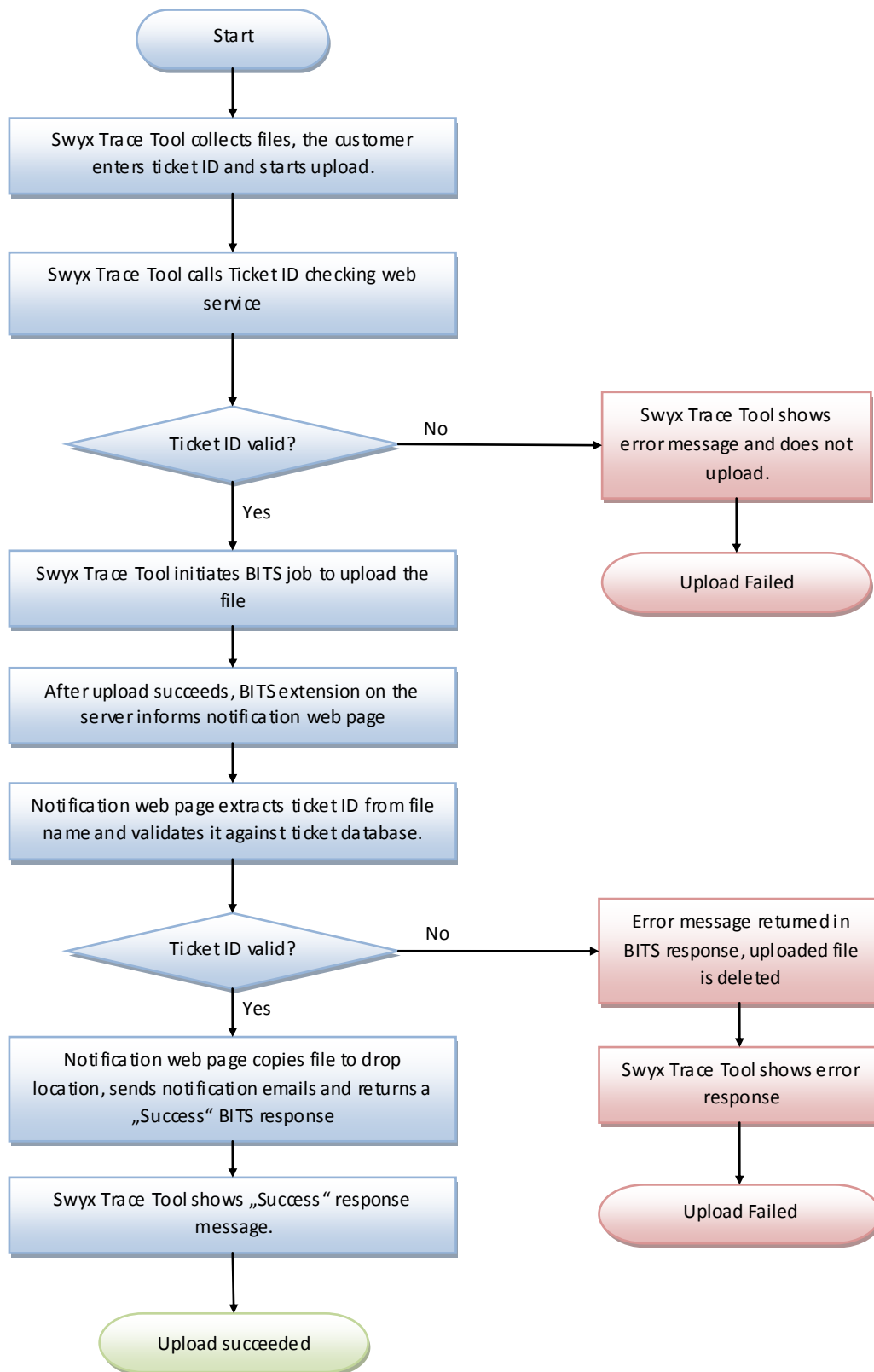
Before uploading trace files Swyx Trace Tool prompts for a support ticket ID. The name of the zip file to upload contains this ID. Optionally Swyx Trace Tool can call a ticket ID checking web service to validate the ID. For uploads to Swyx this check is mandatory. Swyx uses such a web service to check if there's an open support ticket with that ID. If you setup an own upload server you can offer such a validation service, too. Swyx Trace Tool Server ships with a sample web service you can use as basis for own developments. This document describes how this sample is installed.

BITS Server extensions can notify another web application about completed uploads. Such an application can react on uploads, e.g. send an email or move the uploaded file to another location. Swyx Trace Tool Server ships with a sample notification application which sends an email and moves the uploaded file to a separate upload folder. This document describes how to install and configure this sample application.

The notification sample web application and the ticket id checking web service sample application are available with full source code.

Swyx Trace Tool Upload process

The complete Swyx Trace Tool upload process works as follows:



The additional ticket ID check after the upload is a countermeasure against malicious uploads. Upload of arbitrary files using a third-party BITS client only succeed with the correct file name syntax. In that case the file name has to contain a valid ticket ID and the upload does not remain unnoticed. The preliminary ticket ID check Swyx Trace Tool performs via calling the web service before the

upload is a convenience mechanism, because the customer gets an immediate response if he enters an invalid ticket and does not need to wait until the upload took place.

Install and configure BITS Server extensions

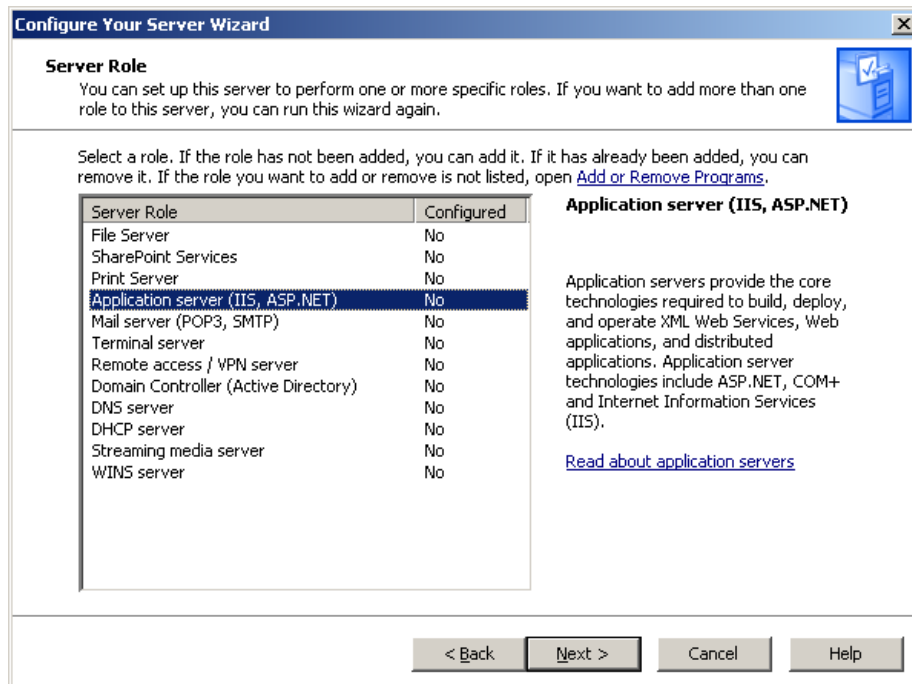
The following instructions assume that you have a Windows Server 2003 without any enabled roles.

You may need your Windows Server 2003 installation CD.

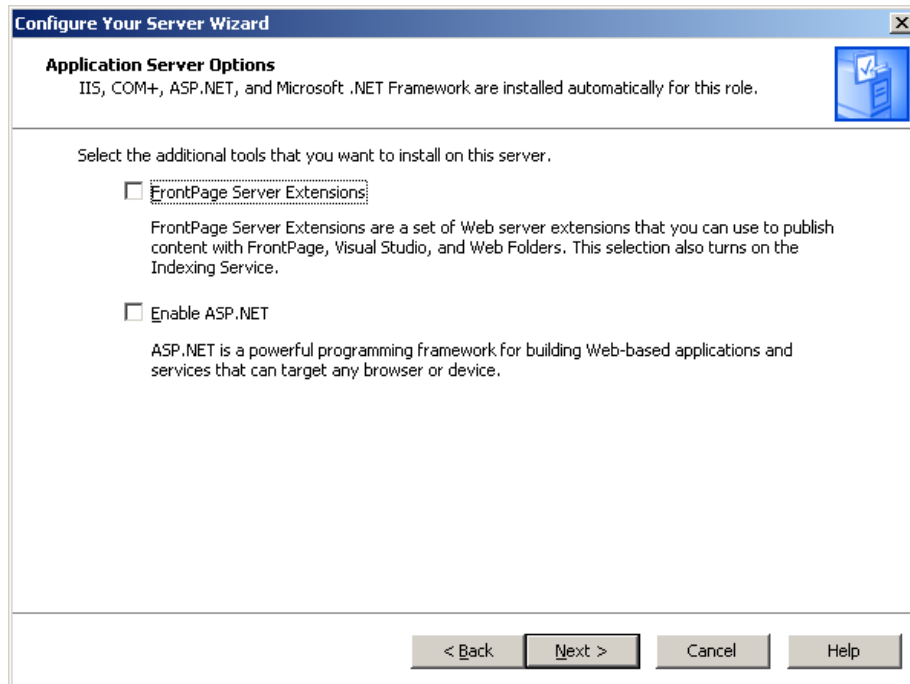
Step 1: Install IIS

In this step Internet Information Server is installed on Windows Server 2003.

1. Start **Server Manager** by selecting **Manage Your Server** from the start menu
2. Click **Add or remove a role**.
3. The **Configure Your Server Wizard is shown**. Click **Next** to show the list of available server roles:



4. Select **Application Server (IIS,ASP.NET)** and click **Next**. The **Application Server Options** dialog is shown:



5. If you plan to implement a support ticket ID checking web service check the option **Enable ASP.NET**. Click **Next** to continue. The **Summary of Selections** dialog is shown. Click **Next** to start the installation. You might need your Windows Server 2003 installation CD.
6. Click **Finish** to close the wizard after the installation is complete.

Step 2: Install BITS Server Extensions

In this step the BITS Server Extensions for IIS are installed.

1. Click **Start** and then click **Control Panel**.
2. Select **Add or Remove programs** and then click **Add or Remove Windows Components**.
3. From the **Windows Components Wizard**, select **Application Server**, and then click **Details**.
4. In the **Application Server** dialog box, select **Internet Information Services (IIS)**, and then click **Details**.
5. In the **Internet Information Services (IIS)** dialog box, check **Background Intelligent Transfer Service (BITS) Server Extensions**.

By default, both the BITS server extension ISAPI and the BITS management console snap-in are installed. Keep both.

6. Close the **IIS** and **Application Server** dialog boxes to return to the **Windows Component Wizard**. Then click **Next** to begin the installation of BITS Server Extensions as part of Application Server. The wizard completes the installation.

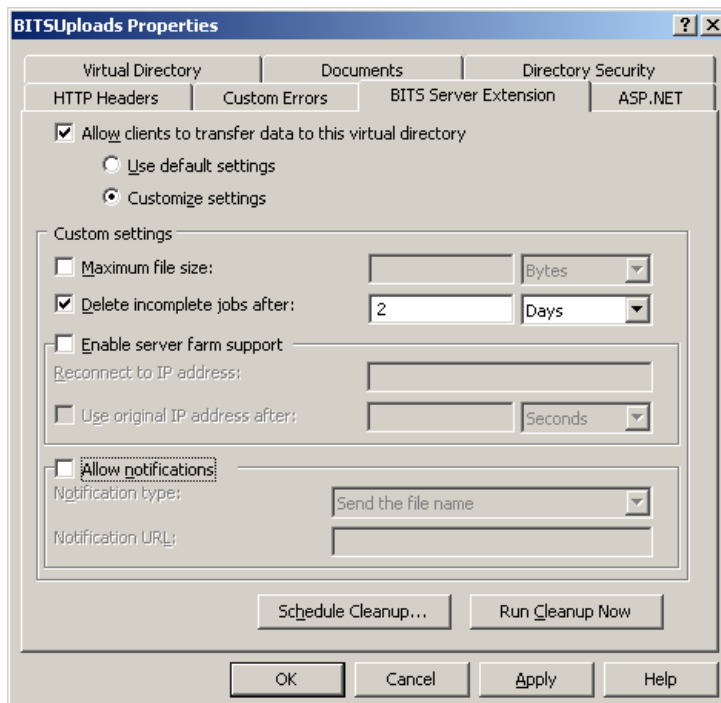
Step 3: Configure BITS

In this step a virtual directory in IIS is created and configured for file uploads via BITS.

1. Start **Server Manager** by selecting **Manage Your Server** from the start menu
2. Click **Manage this application server**.
3. Expand the **Internet Information Services (IIS) Manager** node.
4. Right-click on **Default Web Site** and select **New Virtual Directory**.
5. The **Virtual Directory Creation Wizard** is shown. Click **Next**.
6. Enter an alias for the directory, e.g. **BITSUploads**. Click **Next**.
7. Enter the path for virtual directory content, e.g. "C:\inetpub\wwwroot\BITSUploads". Click **Next**.
8. On the **Virtual Directory Access Permissions** dialog make sure that only **Log Visits** is checked. Click **Next** and click **Finish** to complete the wizard

Now a virtual directory named **BITSUploads** is available. BITS server extensions need to be enabled on this directory.

9. Right-click on the **BITSUploads** virtual directory and select **Properties**. Select the **BITS Server Extension** property page
10. Check **Allow client to transfer data to this virtual directory**.
11. Select option **Customize Settings**.
12. Make sure that **Delete incomplete jobs after:** is checked. Define an appropriate time interval. For uploads from Swyx Trace Tool choose a rather short time frame, e.g. 2 days.

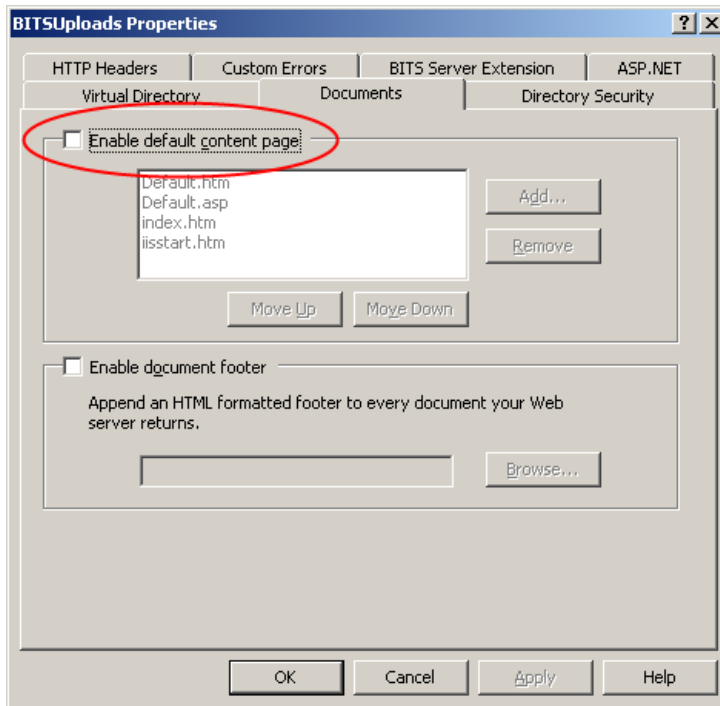


13. Click **Apply**.
14. Click **Schedule Cleanup**.
15. Define how often a directory cleanup should take place. Keep the defaults, if not sure. Click **OK** to confirm the schedule.

This configures the frequency with which BITS Server Extension scans the virtual directory for incomplete upload files.

16. Select the **Documents** tab.

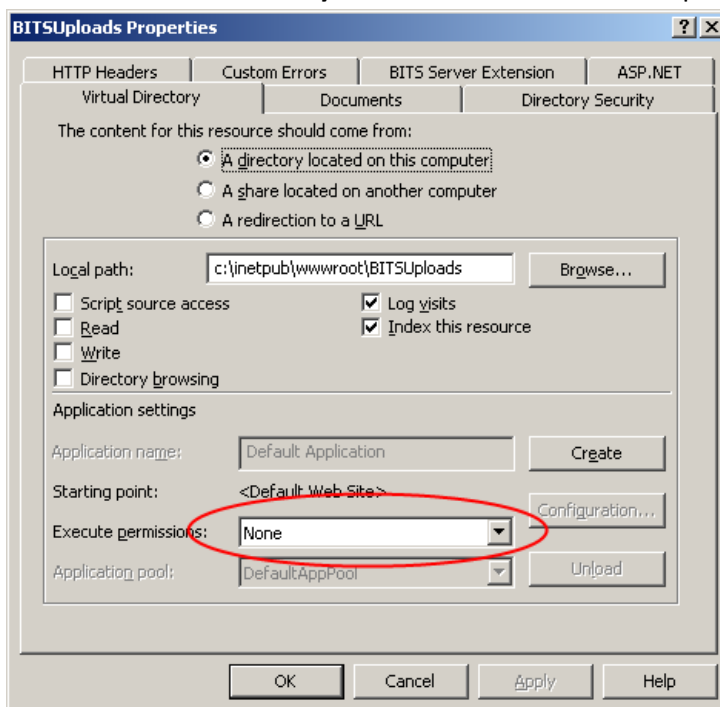
17. **Uncheck Enable default content page:**



18. Select the **Directory Security** tab.

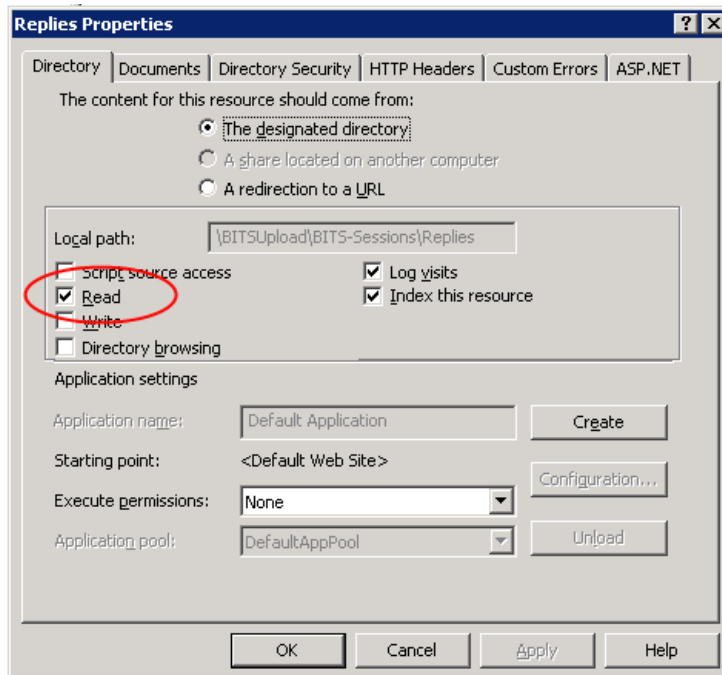
19. Make sure that **Enable anonymous access** is enabled. Remember the Windows user account which is entered here. Typically this is IUSR_Servername, where "Servername" is your Windows Server name.

20. Select the **Virtual Directory** tab. Make sure that Execute permissions is set to **None**:



21. On the same page, make sure that neither **Read**, nor **Write**, nor **Directory Browsing**, nor **Script source access** is checked

22. Click **OK** to close the virtual directory properties.
23. Right click on the **BITS-Sessions\Replies** subfolder of the **BITSUploads** virtual directory and select Properties. Make sure that **Read** is checked:



This is necessary to allow Swyx Trace Tool to read the reply message generated for an upload.

24. Click **OK** to close the properties.

Now the file system permissions have to be adapted so that BITS server extensions are able to store files in the virtual directory.

25. Right-click on virtual directory **BITSUploads** and select **Permissions**.
26. Select the Windows user account used for anonymous access. This is the account you saw in step 19 above.
27. Grant **Modify** permissions and click **OK** to close the dialog.

The server is now configured to receive uploads via BITS. In Swyx Trace Tool use the Upload URL:

<http://yourserver.example.com/bitsuploads>

(Replace yourserver.example.com with your server name)

Step 4: Test BITS Server configuration

To test your upload server configuration you can use bitsadmin.exe, a command-line tool which can be installed from your Windows Server 2003 CD, folder \support\tools. Execute support.msi located in that folder to install it. You find the tool in \program files\support tools.

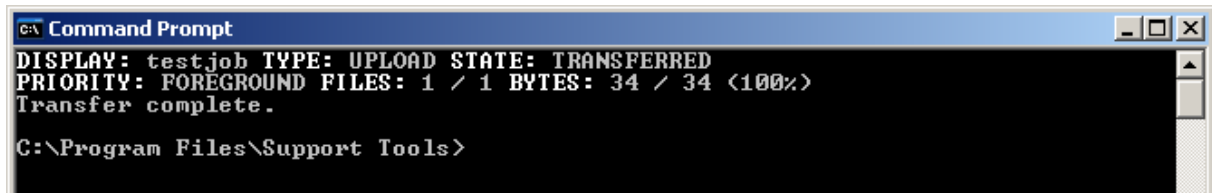
This step assumes that you configured your server with the settings described above. If you have chosen other settings, e.g. a different virtual directory alias or location you have to adapt the instructions in this section.

To upload a file use the following command-line (this is one line)

```
Bitsadmin.exe /transfer testjob /priority foreground /upload
http://servername/bitsuploads/testfile.txt c:\testfile.txt
```

This commandline assumes that a file testfile.txt, located in c:\ is to be uploaded to server with name “servername” and the virtual directory configured for BITS uploads is named BITSUploads.

BITSAdmin shows the current state of the upload. This looks like this:



```
C:\ Command Prompt
DISPLAY: testjob TYPE: UPLOAD STATE: TRANSFERRED
PRIORITY: FOREGROUND FILES: 1 / 1 BYTES: 34 / 34 <100%>
Transfer complete.
C:\Program Files\Support Tools>
```

After it has been completed you should find a file named testfile.txt in c:\inetpub\wwwroot\bitsuploads on the server.

Details about bitsadmin.exe can be found here:

<http://technet2.microsoft.com/windowsserver2008/en/library/4853036e-1df8-45ad-8be6-cfb097b8dd271033.mspx>

Note: Bitsadmin.exe /transfer sometime shows an “Unable to complete job – 0x80200002” error. You can ignore that if the state is TRANSFERRED and you see the uploaded file on the server.

Step 5: Install upload notification sample

BITS Server extension is able to notify a web application when an upload has been successfully completed. Swyx Trace Tool Server ships with a sample ASP.NET application called BITSNotify you can use as basis for own implementations. The sample BITSNotify application checks the ticket ID against a SQLServer database, moves uploaded files into a configurable folder and sends an email notification to inform about the upload.

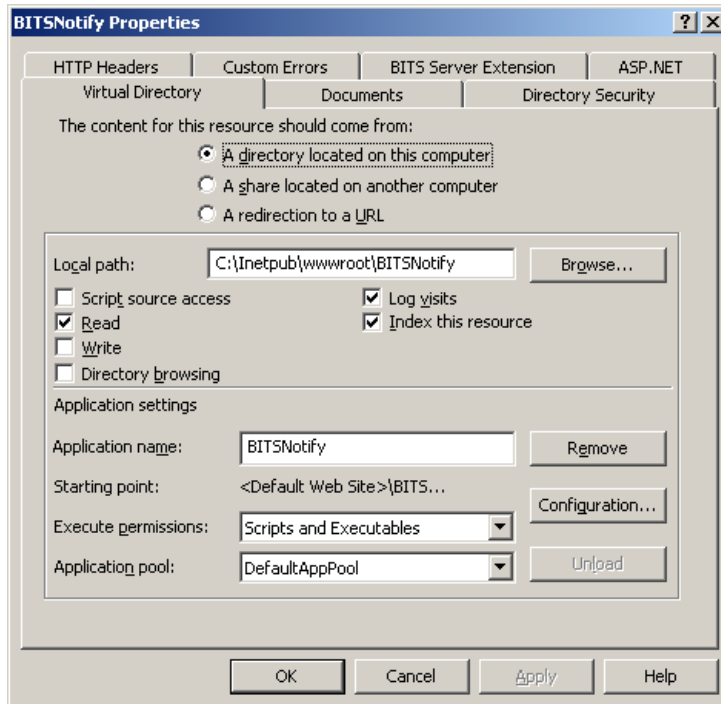
This step shows how to install and configure the application

First create a new virtual directory in IIS called BITSNotify.

1. Open **IIS Manager**, right click on **Default Web Site** and select **New Virtual Directory**
2. The **Virtual Directory Creation Wizard** is shown. Click **Next**.
3. Enter an alias for the virtual directory, e.g. **BITSNotify** and click **Next**.
4. Choose a path for the directory contents, e.g. c:\inetpub\wwwroot\bitsnotify and click **Next**.
5. On the **Virtual Directory Access Permissions** dialog check **Read** and **Execute (such as ISAPI applications or CGI)**
6. Click **Next** and click **Finish** to complete the wizard.

Now configure the BITSNotify directory.

7. Right-click on the **BITSNotify** virtual directory and select **Properties**.
8. An application named BITSNotify should already been configured. Access permissions should already be set to **Read**, **Log visits** and **Index this resource**. The application name is already defined as **BITSNotify**, execute permissions should be **Scripts and Executables** and the **DefaultAppPool** application pool is selected:



9. Select the **Documents** tab, uncheck **Enable default content page**
10. Select the **Directory Security** tab. Make sure that anonymous access is enabled in the **Authentication and access control** settings. In the **IP address and domain name restrictions** settings make sure that access is allowed from 127.0.0.1 only, because this web application will only be called from the BITS server extensions on the same system.
11. Select the **ASP.NET** tab and make sure that **ASP.NET version** is set to **2.0.50727**

Next step is to install and configure the BITSNotify sample application

12. Copy the contents of the Swyx Trace Tool Server **BITSNotify** folder into the virtual directory folder you have set in step 4 above, e.g. c:\inetpub\wwwroot\bitsnotify.
13. Open file **web.config** with notepad or another text editor.
14. Locate the **BITSNotify.Properties.Settings** element. It contains all parameters you can configure.
15. Set **ToAddress**, **SmtptServer** and **FromAddress** for the email notification
16. Set **DropLocation** to define the folder to store uploaded files. Make sure that the BITSNotify web application has write access to that folder. If you kept the default settings above, the

application runs in the **DefaultAppPool** application pool which runs with the **NetworkService** windows account.

17. The notification email contains a hyperlink for the uploaded file. Parameter **DropLocationURL** defines that link. BITSNotify will append the file name use the resulting hyperlink in the email.

If the email recipient should access the uploaded file via HTTP:

Configure a virtual directory for the folder defined in **DropLocation** and use the http: URL to that virtual directory in **DropLocationURL**.

If the email recipient should access the drop location via Windows file share:

Share the drop location folder and use an appropriate file: URL to that share as **DropLocationURL**.

Test the notification web application

If you open <http://127.0.0.1/BITSNotify/notify.aspx> in a web browser on the server you should see

„BITS Notification application ready.“

And a test email should have been send to the configured recipient. If you get a 404 error, ASP.NET 2.0 might not be correctly configured. Try calling

Aspnet_regiis.exe -i

which can be found in c:\windows\microsoft.net\framework\v2.0.50727.

Now BITS Server extensions have to be configured to call the notification web application.

18. Expand the **BITSUploads** virtual directory in IIS Manager and select **Properties**.
19. Select tab **BITS Server Extension**.
20. Check **Allow notifications**, choose notification type **Send the file name** and enter notification URL: <http://127.0.0.1/BITSNotify/notify.aspx>
21. Click **OK** to save the settings.

If you now try an upload again as described in step 4 the uploaded file should be placed in the **DropLocation** you defined and an the recipient configured in **ToAddress** should have got an email.

Step 6: Install and configure the Ticket-ID Checking Web Service

Swyx Trace Tool can be configured to check the ticket ID entered by the customer before the upload. The BITSNotify web application you've installed in the previous step contains a sample webservice.

The sample assumes that the tickets are checked against a Microsoft SQLServer database, e.g. your support ticket system's database.

1. To enable the ticket ID check open web.config in the BITSNotify directory (c:\inetpub\wwwroot\bitsnotify if you used the default location in step 5), locate **BITSNotify.Properties.Settings** and set **ValidateTicketID** to **true**.

Note: If ticket ID checking is enabled the uploaded files must have the following file name syntax:

```
<name>_<ticketid>_<yymmdd>_<hhmmss>.zip
```

Or, more precisely, the name is checked against the following regular expression:

```
^.+_(.)_\d{6}_\d{6}.zip$
```

Additionally the ticket ID will be checked against an arbitrary Microsoft SQLServer database by calling a stored procedure with the following syntax:

```
CREATE PROCEDURE dbo.sp_ValidateTicket
@TicketID varchar(50)
```

It returns 1 if the given ticket id is valid and 0 otherwise. You have to define the database connection to use.

2. To set the database connection string locate the **<connectionString>** element in **web.config** and set the element named **TicketDBConnection** to point to your SQLServer database, e.g.

```
Data Source=myServerAddress;Initial Catalog=myDataBase;User
Id=myUsername;Password=myPassword;
```

Note: You should encrypt the connection string. See [http://msdn2.microsoft.com/en-us/library/dx0f3cf2\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/dx0f3cf2(VS.80).aspx) for details.

To check if it works open the following url with a web browser on the server:

<http://localhost/BITSNotify/uploadrequest.asmx>

You should see something like this:

