



iService Technical Guide

© 2025 One-to-One Service.com, Inc.

This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

1. Introduction	5
2. Installation And Upgrades	11
2.1 Initial Installation	12
2.1.1 The Release Archive	14
2.1.2 Setup the Web/Application Windows Server	17
2.1.3 Run iServiceSetup.exe	24
2.1.4 Create Initial Database	27
2.1.5 Create Tenants	29
2.1.6 Install Website	31
2.1.7 Install Reports	43
2.2 Installing Upgrades (v8 and later)	51
2.3 Upgrading Reports	64
3. Technical Configuration and Tools	65
3.1 Management Console	66
3.2 Mailpopper Monitor	81
4. Maintenance Tasks	85
4.1 Purging Interactions	86
4.2 Purging Contacts	87
4.3 Moving Raw Messages Table	87
5. Troubleshooting	93
5.1 Unable to Generate a Temporary Class	94
5.2 Messages Not Leaving iService	95
5.3 Checking table sizes in iService	96
5.4 iService Error Log Files	97
5.5 Incorrect URL for Web Services	97
Index	99

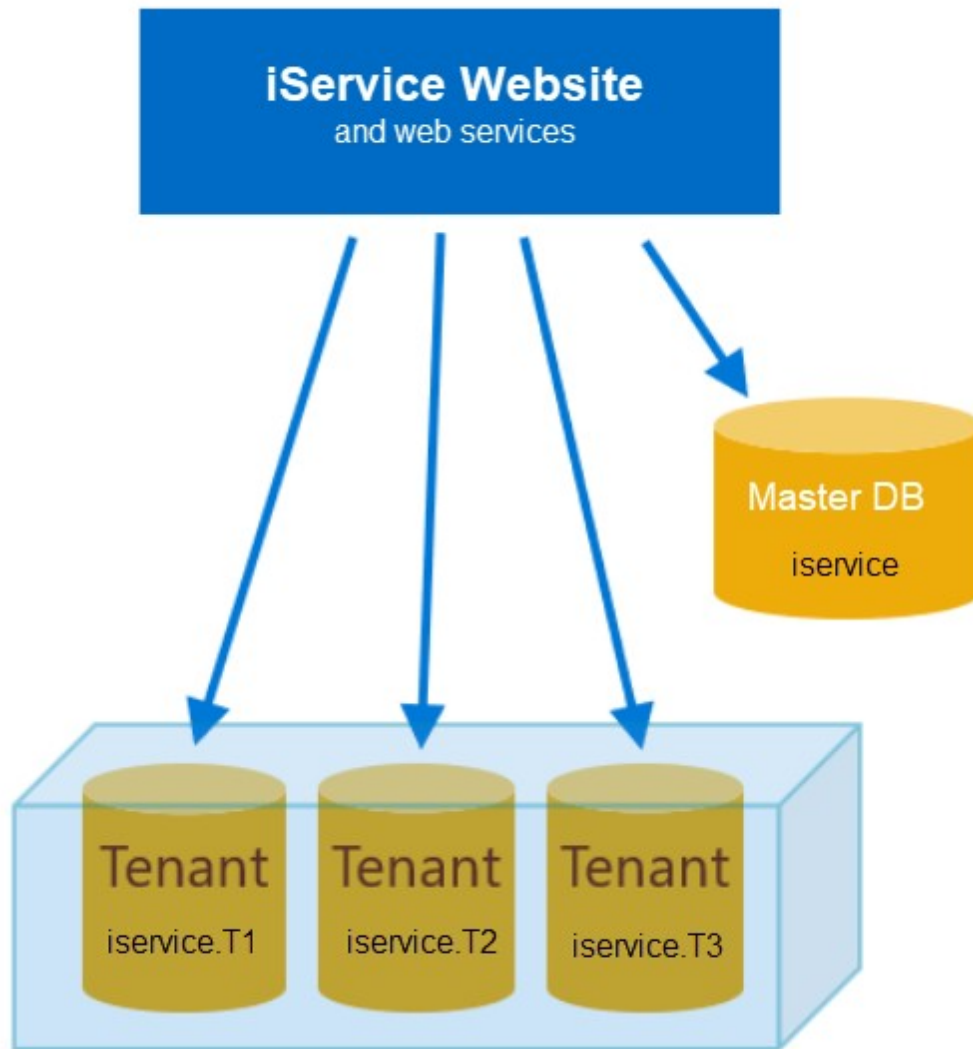
This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

Introduction

1 Introduction

This user guide is intended for network administrators responsible for the installation, upgrade, and technical configuration of an iService installation. Most of the set up and configuration of an iService tenant is done through the web interface. See the [iService user guide](#) for details on configuring a tenant using the web interface.

iService is a multi-tenant application that uses a single app but separate databases for each tenant. This simplifies the upgrade process because there's only one website and set of web services for the installation. The schemas for each tenant database are identical, and are updated automatically when new versions of iService are installed. The configuration for each tenant, including custom forms and changes to the user interface, are managed through settings within the tenant's database.



iService uses a single website and separate tenant DBs

iService can be installed on a single server, or multiple servers depending upon the number of users within your installation. It is a .Net application that requires a Microsoft Internet information service (IIS) web server and a Microsoft SQL Server database. This user guide does not make any distinction between a single server or multiple server configuration.

The components of iService are briefly described below.

SQL Server Database

- iService stores all of its information for the tenants and the overall installation within a set of Microsoft SQL Server databases. Microsoft SQL Server 2016 or newer are certified with iService.
- The reports used with iService business intelligence are built on Microsoft SQL Server reporting services (SSRS). Although SSRS can be run on your primary database server, for security purposes it is recommended that SSRS operate on a separate server. This is because the SSRS server is directly exposed to users and has a larger attack surface than the SQL Server database, which can be located on internal IPs in a more secure location. Reports are only compatible with SSRS 2016 and newer. The SSRS Web Services should operate on a separate server, but the SSRS database may reside on the database server.

iService Website

NOTE: The IIS Web Server must be running at least .NET 4.6.2.

The installer will detect your version of .NET and prompt for updating if you are not running the correct version. If you are running other applications on the web server, ensure the new version is compatible with your other applications.

- Users configure their iService tenant and access all of its features through the iService website. The website requires Microsoft IIS 8 or higher.
- iService is built upon a set of Web services that interact with the iService database. The Web services file (iService.asmx) is typically located on the same Web server as the iService website, but can operate on a separate Web server if desired.
- Any error log files generated by the web services will be written into the folder that contains the web services. You must ensure that appropriate access is granted for writing to this folder. The deploy script will set permission automatically.

Mailpopper Service

NOTE: The server running the Windows service must be running at least .NET 4.6.2.

- iService uses a Windows service that gathers mail from monitored mailboxes, launches mail processing threads, and performs a variety of other functions. This service can be installed on any Windows server (we do not recommend running the mailpopper service on the database server).
- The MailPopper service interacts heavily with the iService Web services, and the best performance is usually found by installing the service on your Web server that contains the iService Web services file.
- The DLLs for this service are by default stored in c:\Program Files\1to1Service\iService. Any error log files generated by the mail popper service will be written to this folder.

This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

Installation And Upgrades

2 Installation And Upgrades

Upgrades to iService are performed using the same installer files as a new installation. The primary difference is that your initial installation process requires you to create a master database that contains information for all of your tenants. When applying upgrades you will skip the Create New Master Database process.

2.1 Initial Installation

You must have administrator access to the servers on which you will be installing iService. You will also need the SQL user that has the create database permission to create the iService database. For security purposes, it is best to have a new SQL account created by your DBA for this purpose rather than use the default SA user account shipped with SQL Server. This installation guide assumes you have already installed MS SQL Server on your database server, and MS IIS on your web server.

Server Settings

The iService Web Application must be installed on a Windows Server running IIS 8 or newer. We recommend enabling websockets on your server. See

<https://www.iis.net/configreference/system.webserver/websocket>.

- The web server must have the .NET 4.6.2 or higher run-time package installed (If it is not preset during installation, iService will automatically download and install it during the setup process). The Web Application pool used by iService needs to be set to use .NET CLR 4.0.
 - If you are using Windows Server 2012, you must add support for .woff2 to your website manually by adding it to the MIME types at the Server Level for IIS.

You can use the IIS command line tool `appcmd` to add it as shown below:

```
c:\windows\system32\inetsrv\appcmd
    set config /
        section:staticContent /+
            "[fileExtension=' .woff2 ',mimeType='
```

- The iService Mail Popper Service should be installed on a Windows 2012 or newer Windows Server, and also requires that the .NET 4.6.2 or higher run-time package be installed.
- The Master and Tenant databases require a SQL Server 2016 or newer database server. Installation requirements for SQL Server are outside the scope of this document. The server must be configured to allow Windows and SQL Server Authentication. This is because you will create a SQL Server User that the iService web services and mail popper will use to access the iService database. It is recommended that you install the SQL Server Reporting Services (SSRS) when you setup your SQL server (assuming SQL Server 2016 or newer) as this is the engine that iService Reporting will utilize. For best security practices, it is recommended to run SSRS on a separate server from your actual iService database. If you are using a version of SQL Server prior to 2016, you must download the SSRS 2016 installer for reports.

When determining what servers to use for the installation of each component, keep in mind the following requirements:

- The Web Server must be reachable by your agents and must be able to open a network connection to your SQL server. This server must also be able to access the outgoing mail server via SMTP for delivery of agent responses. If you plan on utilizing Secure Messaging to your customers or the Knowledge Base, your customers will also need to be able see web pages served by this server.
- The iService Mail Popper Service must be able to access the mail server where incoming messages may be retrieved via POP3 or IMAP (secure POP and IMAP are supported and recommended). This server must also be able to access the outgoing mail server via SMTP for delivery of agent and alert notifications. These services must also be able to open a network connection to your SQL server.
- The SQL server must be reachable by both the Web server and the Mail Popper Service.

2.1.1 The Release Archive

The installation files and installer are included within an iService Release archive that typically has a name such as:

iServiceCRM-8.0.444-2020.01.16-151658.zip

The naming convention is as follow:








8 - the major version number

0 - the minor version number

444 - the build number






2020.01.16 - the build date

151658 - the build time (15:16:58 in 24 hour time)

Name	Type
 Binaries	File folder
 Deploy Tools	File folder
 For New Installation	File folder
 iService Utilities	File folder
 Reports	File folder
 Website	File folder
 iServiceSetup.exe	Application

Contents of the Release archive



Deploy tools

Name	Type
 ArchivedDeploys	File folder
 iservicesetupfiles	File folder
 deployiservice.bat	Windows Batch File
 deployiservice.cfg	CFG File
 wtee.exe	Application

Contents of the Deploy Tools Folder

The deploy tools folder contains the files needed for the iService installation scripts.

For New Installation






Name	Type
 iService.MailPopper.config	CONFIG File
 Web.config	CONFIG File

For New Installations Folder

This folder contains configuration files that generally do not change during an update. You will configure them when you first install iService, but do not modify them during release updates unless specifically instructed to make changes.

- Web.config - This is the configuration file used by the iService web services to connect to the iService database. It is located in the root of the iService website.
- iService.MailPopper.config - This is the configuration file used by the iService mail popper to connect to the iService database. It is located in the folder that contains the mailpopper and management console code (usually C:\Program Files\1To1Service\iService).




iService Utilities

Name	Type
 BatchFormSubmit.exe	Application
 BatchUtility.exe	Application
 InteractionCreate.exe	Application
 SampleProxy.dll	Application extension
 Shared.dll	Application extension

iService Utilities Folder

These files should be distributed to users that need to run the iService utilities for processing batch files.

Reports

Name	Type
 iServiceCRM	File folder
 Deploy-iServiceReports.ps1	Windows PowerShell Script
 iServiceCRM.sln	Microsoft Visual Studio Solution

Contents of the reports installation folder

This folder contains the reports installation script and files required to deploy reports to your SSRS server. For details on using the deploy PowerShell script, see the [Install Reports section](#)⁴³.


Website

This folder contains the iService website files to be installed on the web server. If you are performing an update, these files are used to replace your existing website. It does not contain a web.config file since that file contains credentials unique to your installation. When performing upgrades be sure to copy over your existing web.config file.

iServiceSetup.exe

The .exe file is the file that the deploy scripts will use to install iService. You will NOT run this installer directly. Instead, you will run the installer scripts, and those scripts will launch this ext at the appropriate point of the installation.

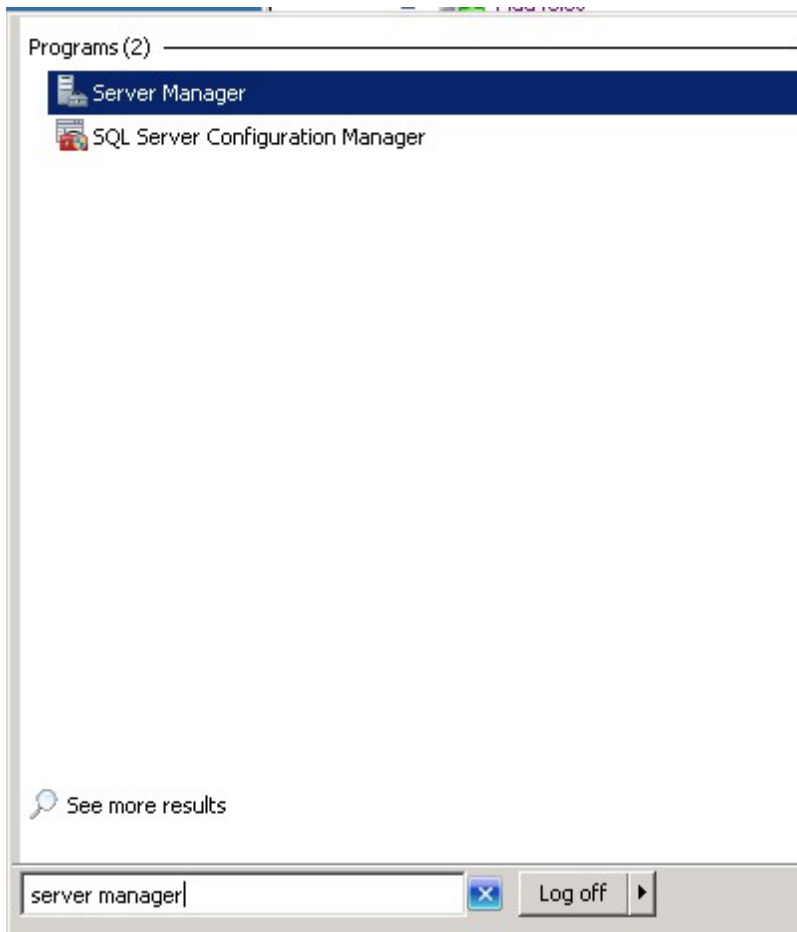
2.1.2 Setup the Web/Application Windows Server

In order for iService to function as a web application on a Windows server, your server must be setup as both an Application Server and as an IIS Web Server. This can be accomplished by adding these roles to the set of roles your server performs. **If your server already has these roles installed, then you may safely skip this section and proceed to “[Install Website](#)”** 

Adding a new role

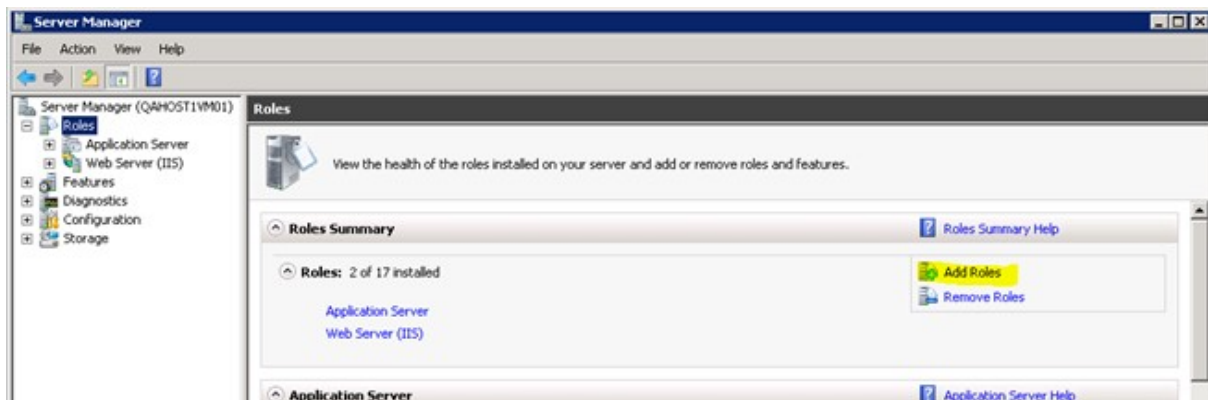
To add new roles, you need to first log into your server as an Administrator. This need not be the actual Administrator account, but you may see fewer security confirmation dialog boxes if you are. However, Roles require administrative rights to be installed.

Run the Server Manager application to pull up the Role Management tools. This is most commonly accessed from the toolbar at the bottom of the server screen as the computer icon next to the Start button, but if this icon is not present, you can find it by using the Start button and typing in Server Manager in the “Search Programs and Files” field as shown here:



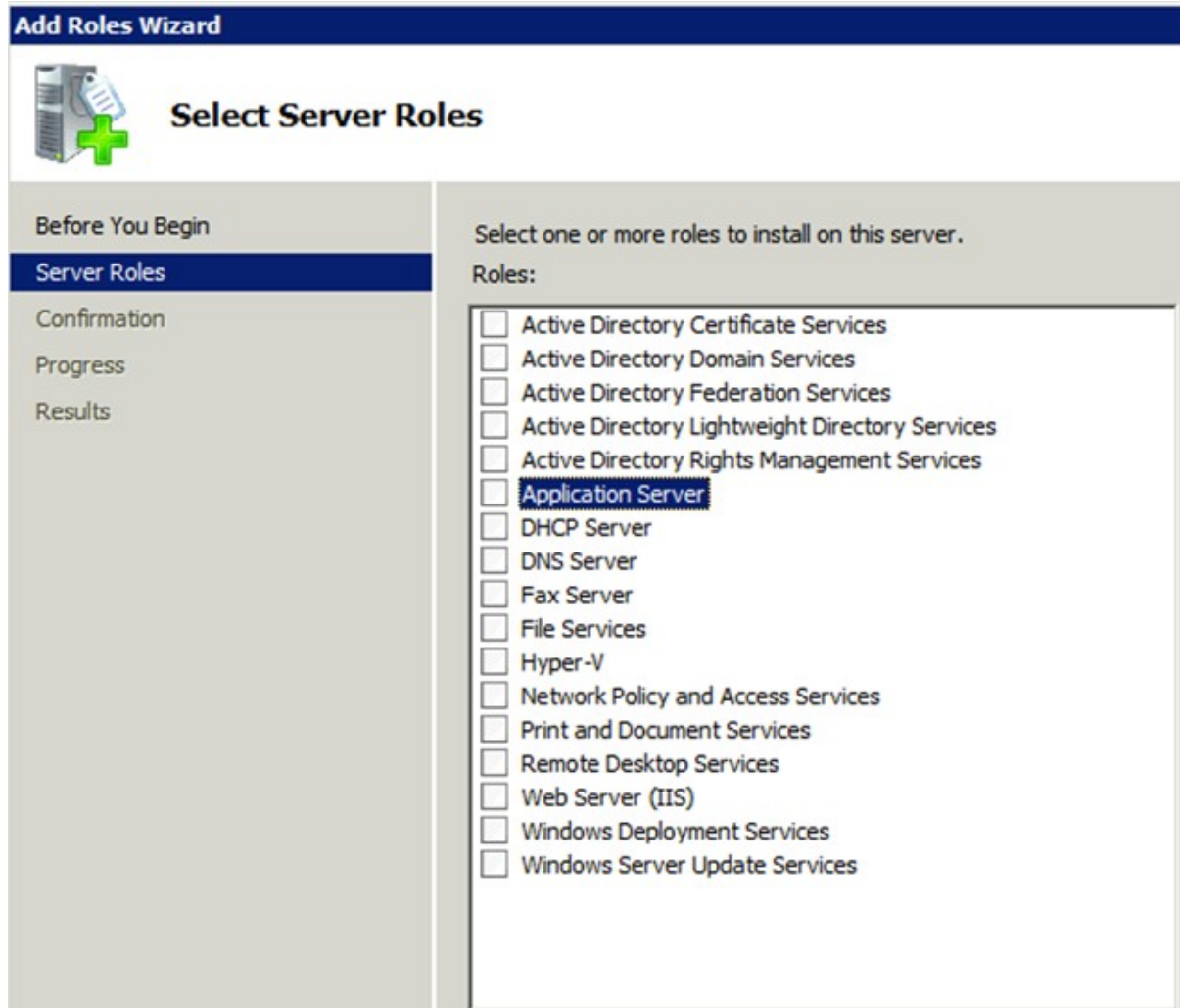
Server Manager

Once the Server Manager is running, select the Roles item in the left pane and then click on Add Roles, which will appear in the far right of the newly opened pane describing existing roles.



Add Roles

Once you've clicked on this, you will be presented with a list of possible roles which can be installed. Select Application Server and Web Server (IIS).



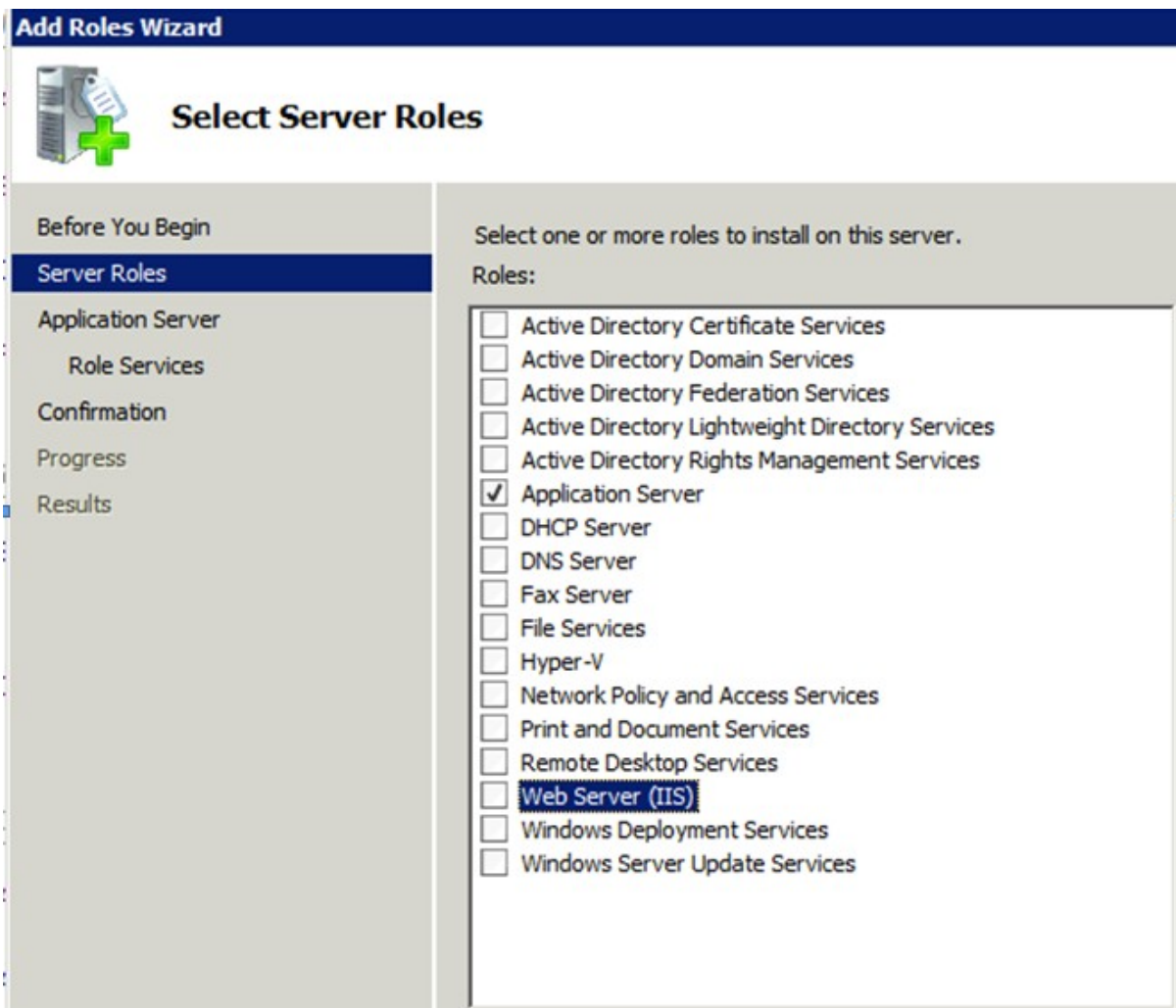
Select Server Roles

When you select Application Server, you will likely be presented with a dialog listing additional features required for the role to perform properly. Click on the Add Required Features button, as shown here:



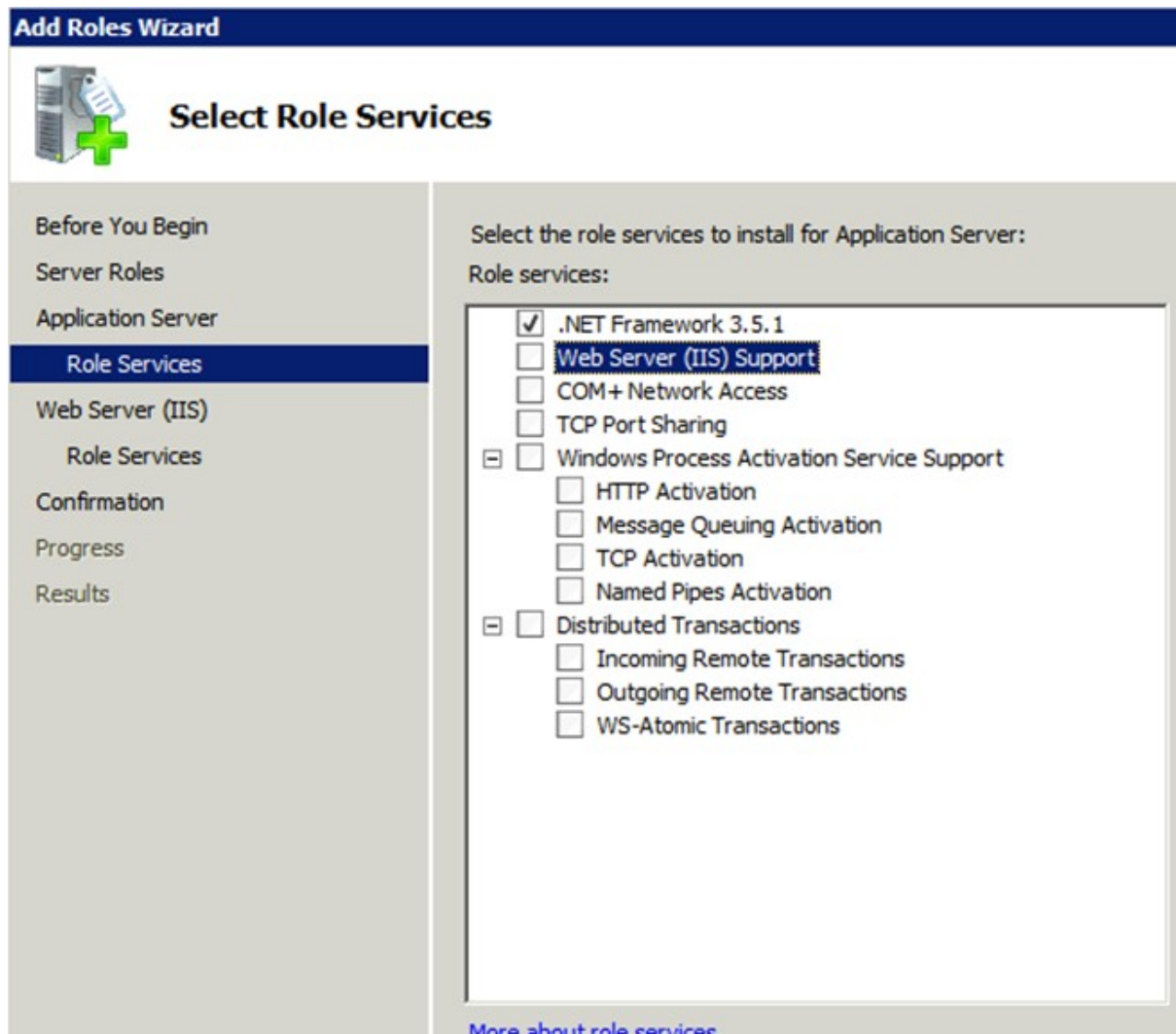
Add Features

Now, add the Web Server IIS role and check it, as shown here:



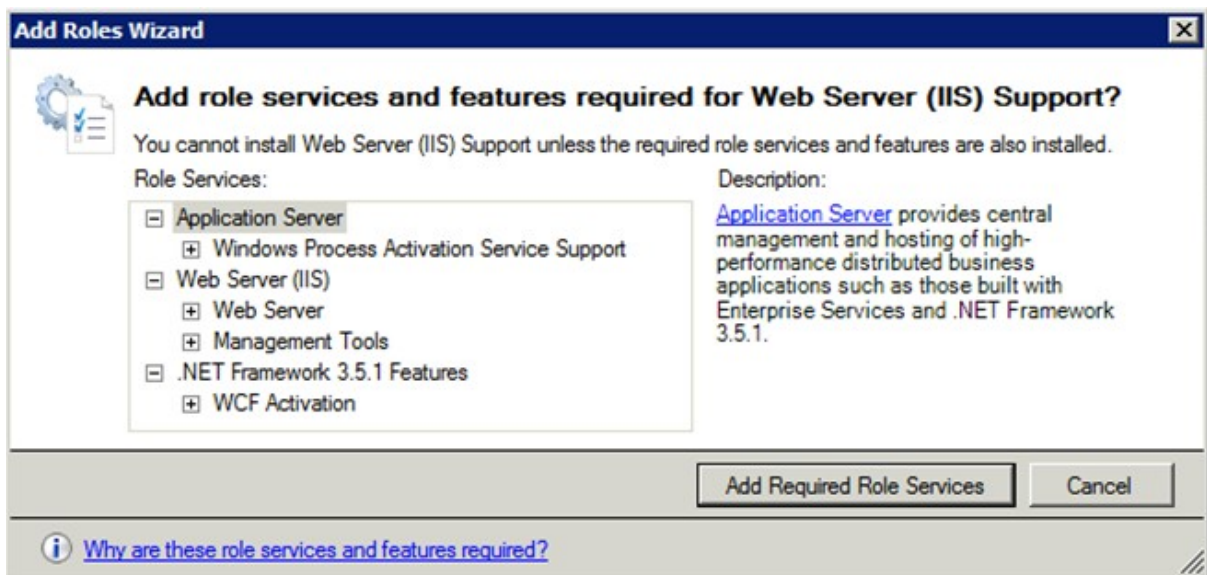
Add Web Server

When you check the box for Web Server (IIS), you will be asked about additional settings which are required for the Roles to be installed as you require them. Select the Web Server (IIS) Support feature for the Application Server roles to ensure proper coordination between the two roles.



Select Role Services

Check this box and then Next to continue.



Add role services and features required for Web Server

These additional features are required for IIS to function properly. Click on Add Required Role Features and then click Next. You will be presented with a confirmation dialog confirming that you wish to install the selected items. Proceed, and your server will be updated to provide the required services.

Once the web server is configured, you should install the IIS redirects module are described in the link below. This will redirect any http requests for your iService website to https.

<https://www.iis.net/downloads/microsoft/url-rewrite>

Use the installer from that page as shown below.

Download URL Rewrite Module 2.1

- English: [Web Platform Installer \(WebPI\) / x86 installer](#) / [x64 installer](#)

2.1.3 Run iServiceSetup.exe

The iService installation program is located in the root folder of the installer archive.

Name	Type
Binaries	File folder
Deploy Tools	File folder
For New Installation	File folder
iService Utilities	File folder
Reports	File folder
Website	File folder
iServiceSetup.exe	Application

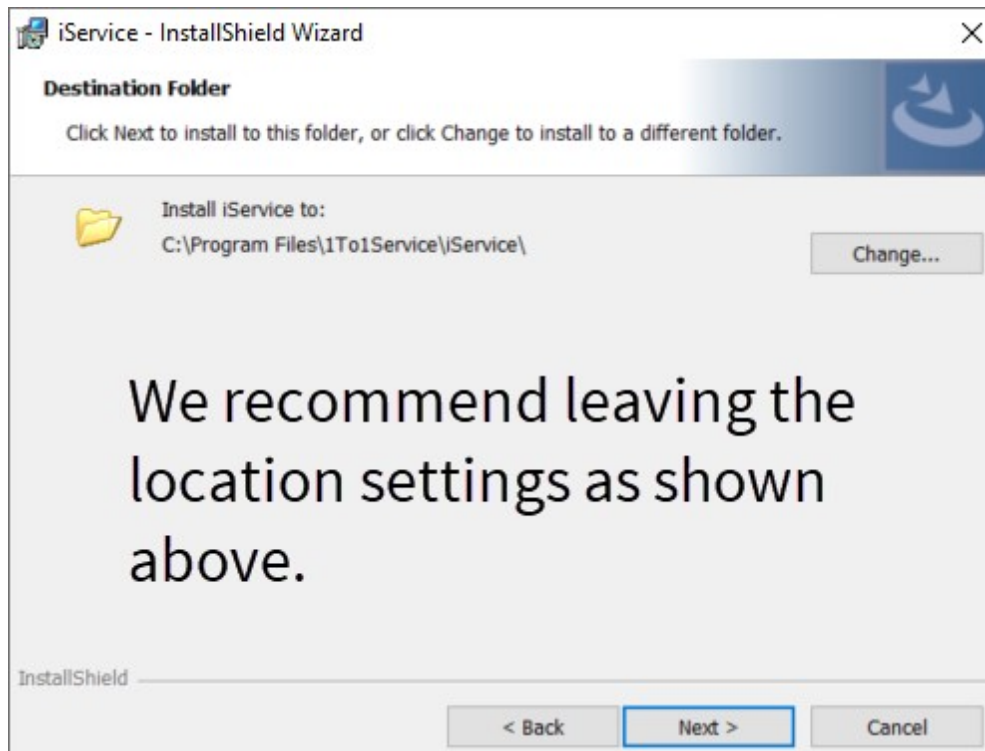
The iService installer

Copy the installer file to web server and run as Administrator. The installer will load as shown below.

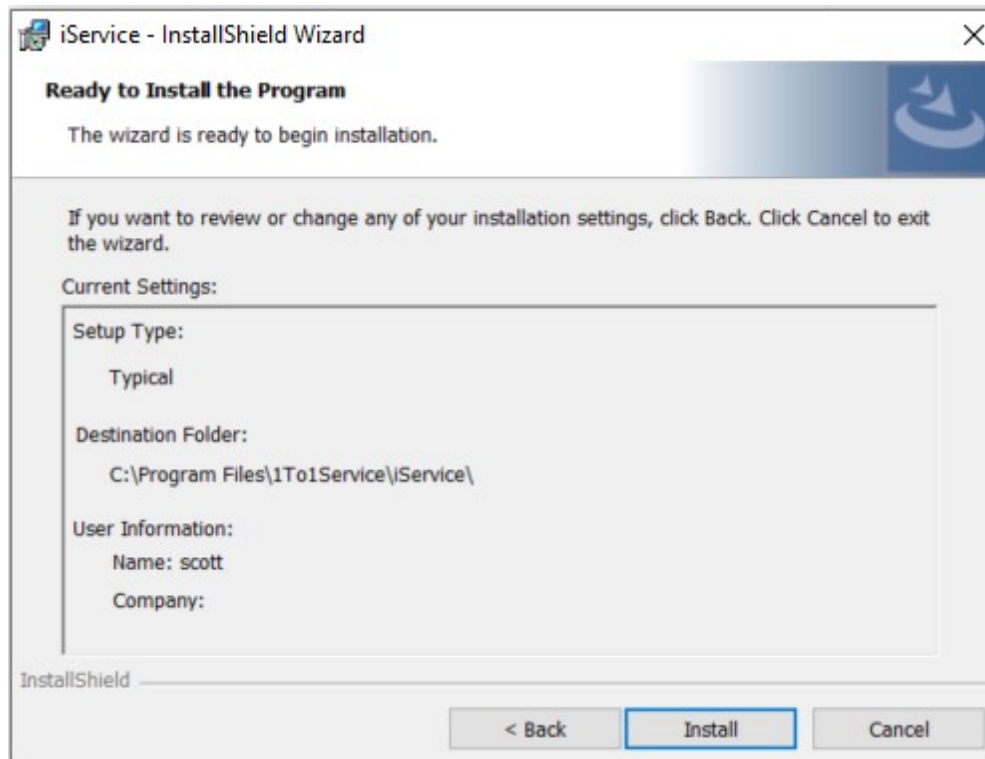


Running the installer

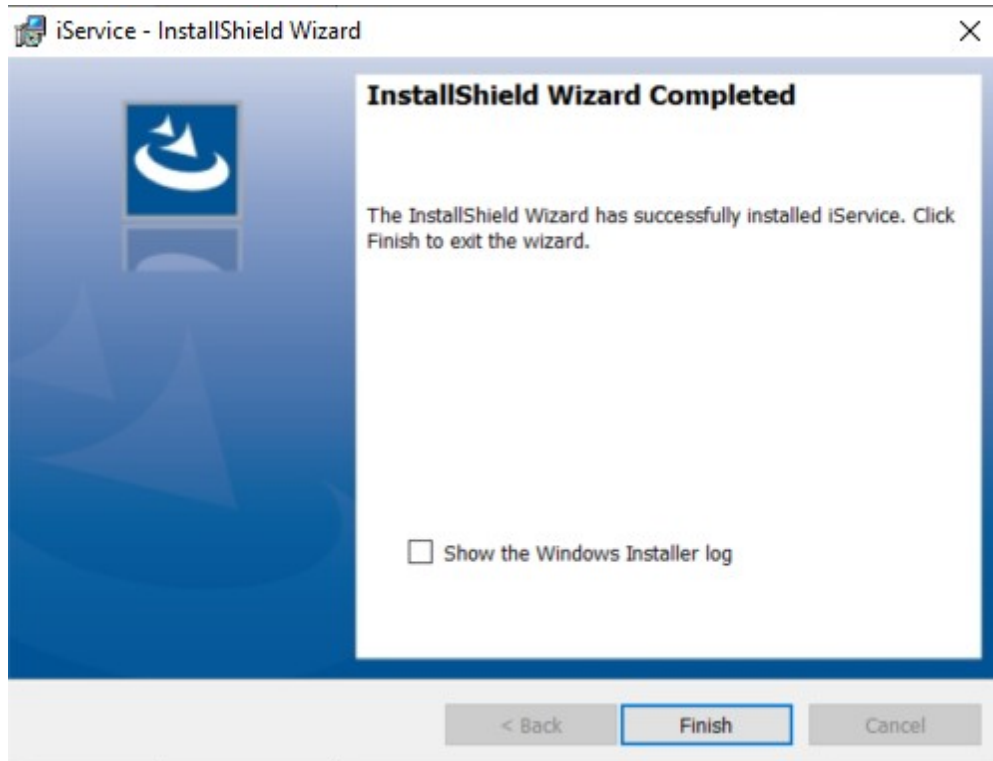
Click next to begin the installation.



Click Next to continue.



Click install to complete the installation.



The completed installation screen

After the installer completes, the management console application will be available in the directory specified above. The directory will look like the below after installation.

This PC > Windows (C:) > Program Files > 1To1Service > iService

Name	Type
iServiceManagementConsole.exe	Application
OneToOne.iService.MailProcessor.Monitor.exe	Application
OneToOne.iService.MailProcessor.Service.exe	Application
AngleSharp.dll	Application extension
FM.dll	Application extension
FM.Server.dll	Application extension
FM.WebSync.dll	Application extension
FM.WebSync.Server.dll	Application extension
HtmlSanitizer.dll	Application extension
MailBee.NET.4.dll	Application extension
OneToOne.iService.Core.dll	Application extension
OneToOne.iService.MailProcessor.dll	Application extension
OneToOne.Shared.dll	Application extension
SampleProxy.dll	Application extension
Shared.dll	Application extension
System.IdentityModel.Tokens.Jwt.dll	Application extension
iServiceManagementConsole.exe.config	CONFIG File
OneToOne.iService.MailProcessor.Monitor.exe.config	CONFIG File
OneToOne.iService.MailProcessor.Service.exe.config	CONFIG File

The management console

The mail monitor

The Windows Service
will not be run as an
application.

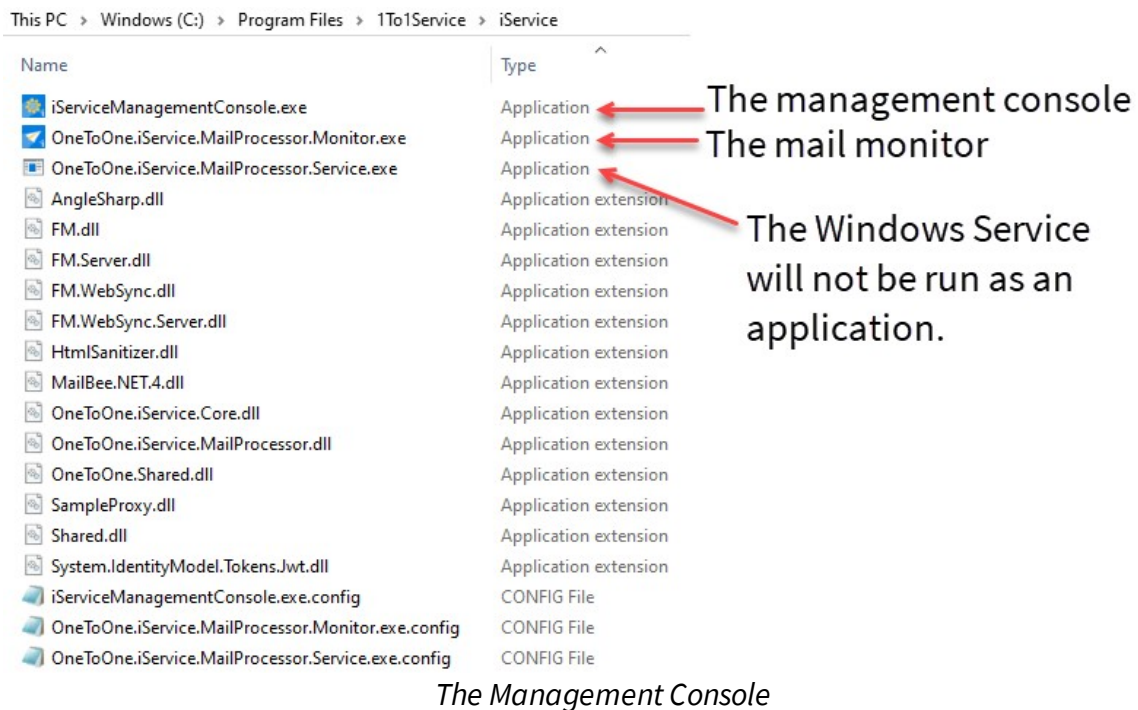
Files in the iService directory after installation

The [Management Console](#)^[66] and [MailPopper Monitor](#)^[81] are Windows applications you will run to configure and monitor your tenant. The MailProcessor Service is a Windows service that will be configured to run upon Windows startup.

2.1.4 Create Initial Database

By default, the iService Windows service, management console, and mailpopper monitor are installed to c:\Program Files\1to1service\iService. Inside this folder you will find the Management Console application used to create your initial master database and tenants.

Launch the iServiceManagementConsole as Administrator to begin the process for creating these databases.



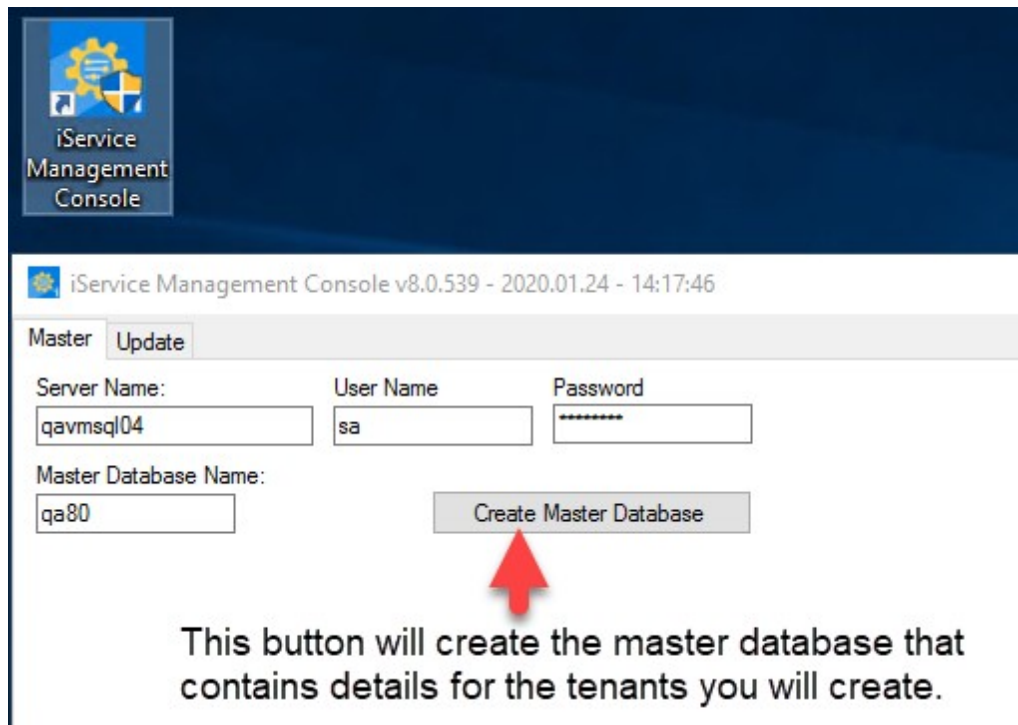
The iService master database contains reference information for all tenant databases. The Master tab includes the following fields, and requires a SQL Server login with DB Owner privileges to the database. This level of access is required because the release updates must have the rights to modify the database.

Server Name - The location of the database server that will contain the iService master database and all of its tenant databases.


User Name / Password - The user name and password within SQL Server required to login to the iService database.

Master Database Name - The name that will be used for the iService master database.

Create Master Database - This button is used to create the initial master database and can also recreate a new version for testing purposes. **If you click this button after the initial installation, it will delete your database and should only be used during initial installation.**



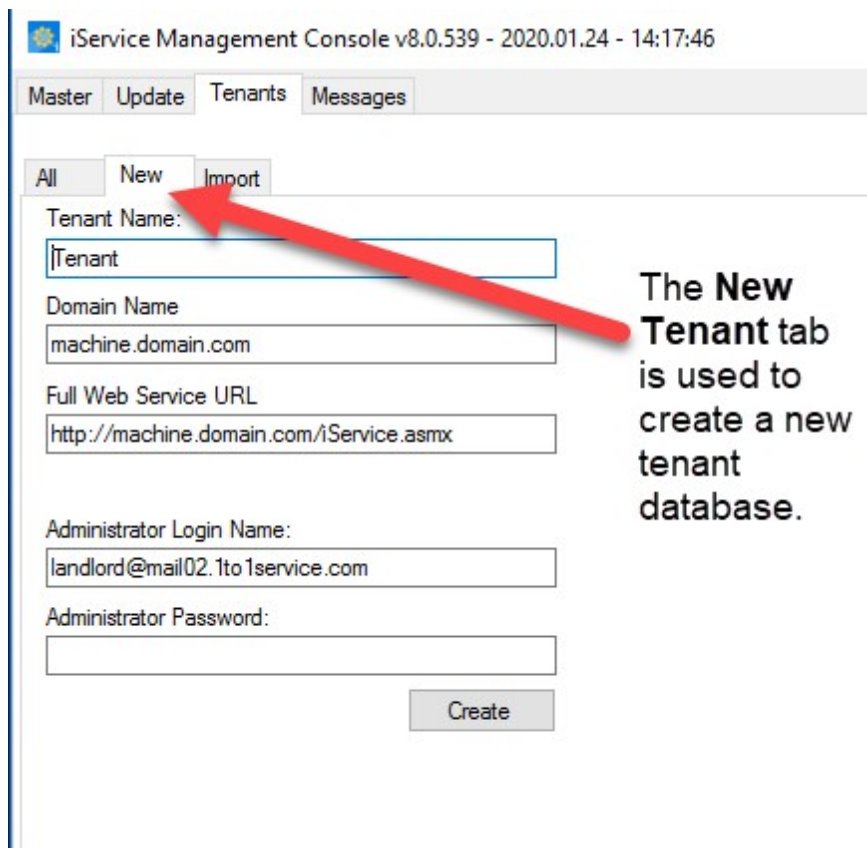
iService Management Console - Create Master Database button

After you successfully create your master database, click to the Update tab. The console will check for database updates, and then then present the Tenants tab. The next step in configuring your installation is to [create your iService tenants](#) .

2.1.5 Create Tenants

The first step in using the iService application is to create a tenant database that can be configured using the iService web interface. The Tenants tab provides a list of all tenants configured, and includes a New Tenant button for creating a new tenant and an Import button for importing tenants from other master databases.

Click New Tenant.



iService Management Console v8.0.539 - 2020.01.24 - 14:17:46

Master Update **Tenants** Messages

All **New** Import

Tenant Name:

Domain Name

Full Web Service URL

Administrator Login Name:

Administrator Password:

Create

The New Tenant tab is used to create a new tenant database.

The New Tenant tab in the management console

To create a new tenant, basic information regarding the tenant is entered into the Landlord Administration component, including the following:

Tenant Name – This must be a unique name, and will be used for the tenant database name. The naming convention for tenant databases is

`masterDBname.tenantDBname.`

Domain Name – The Domain Name specifies the domain where users of the iService system will login and must be unique for each tenant.

Full Web Service URL – This is the location of the web services file for the new tenant. It should typically be the domain name with http or https (as appropriate) followed by /iService.asmx (the name of the web services file).

Administrator Login Name – Enter a login name for the initial superuser of the iService website. This will be contactID = 1 within the new tenant.

Administrator Password – Enter a password for your initial superuser.

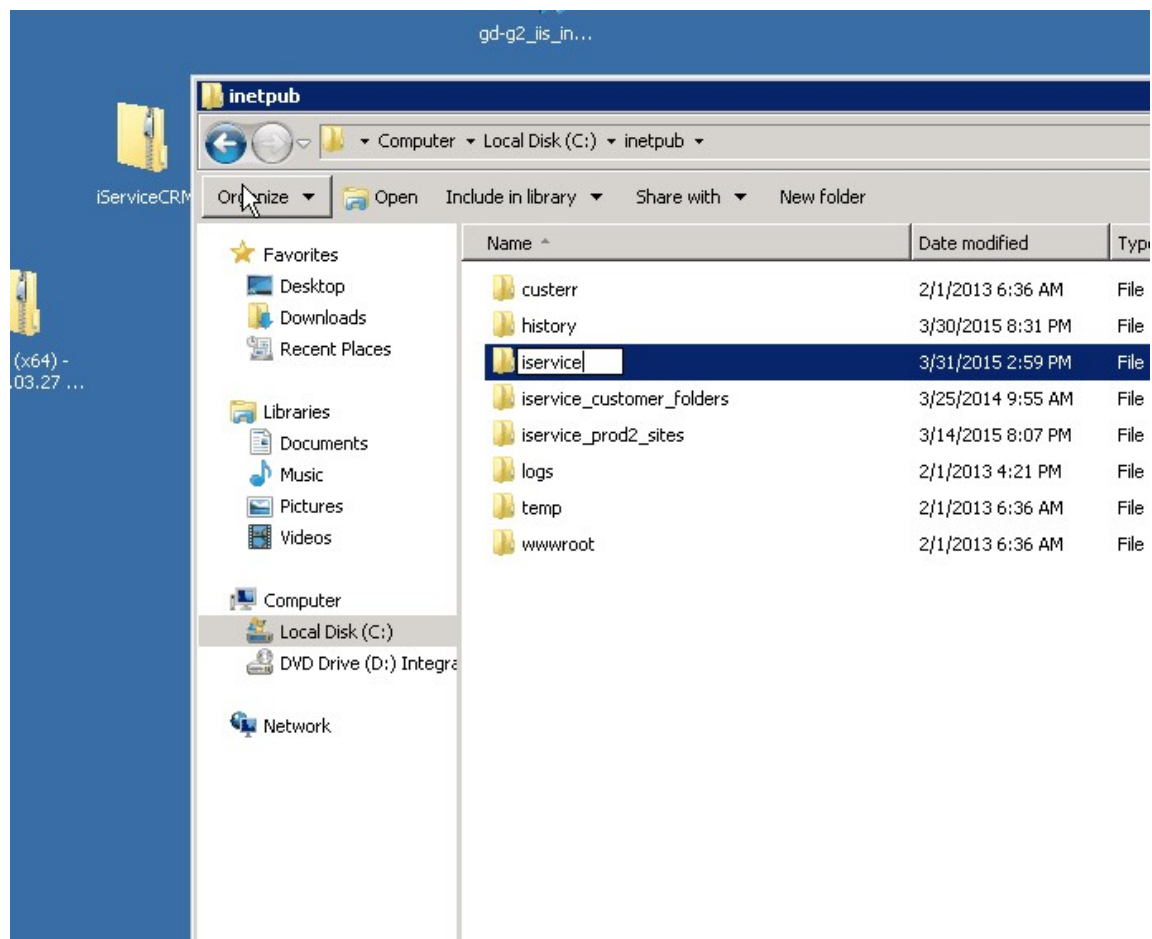
Click the Create button to create your tenant database. Once the database is created and your DNS is setup, you should be able to load the iService website and login using this administrator user ID.

2.1.6 Install Website

There are four steps to setting up your iService website: copy website files, setup web.config, setup mailpopper.config, and configure IIS.

Step 1: Copy files to the web server

To install the web application, you must determine where to store the web site on your web server. By convention, web sites are usually installed in the C:\Inetpub directory on the web server itself. In our example, we create a new folder named iService at this location.



Create website folder

archive and must be copied into the root of the iService website folder you just created.

Edit the web.config to specify your connection details.

appSettings - Update the config file with your connection details for your server and database.

```
<appSettings>
  <add key="databaseServerName" value="servername" />
  <add key="databaseDatabaseName" value="masterDB" />
  <add key="databaseUserName" value="username" />
  <add key="databasePassword" value="userpassword" />
  <add key="asmxUrl" value="https://machine.domain.com/asmx" />
  <add key="aspnet:UseTaskFriendlySynchronization" value="false" />
</appSettings>
```

Server Name - This is the name or IP address of the SQL Server.

Database Name - This is the master database name entered when you created the initial database in the management console.

User Name - This is the SQL user name you specified when creating the initial database.

Password - This is the SQL user password you specified when creating the initial database.

ASM X URL - This is the path to your iservice.asmx file. It was entered when you created your initial tenant, but any valid path to the asmx file will work.

Time Zone Settings

You can modify your web.config file to display your server time zone in a friendly format.

Find this line in web.config:

```
<add key="timeZoneNames" value="(-06:00)=(CST)|(-05:00)=(CDT)"/>
```

The value tag contains a list of strings that are searched and replaced, the search is on the left side of the = and the replacement on the right side of the =. The list is separated by |

So the default searches for 2 strings: -06:00 and -05:00, and replaces them with CST and CDT, respectively. To modify, the user can edit these search/replaces values.

Some examples:

Eastern time:

```
<add key="timeZoneNames" value="(-05:00)=(EST)|(-04:00)=(EDT)"/>
```

Mountain time:

```
<add key="timeZoneNames" value="(-07:00)=(MST)|(-06:00)=(MDT)"/>
```

Pacific time:

```
<add key="timeZoneNames" value="(-08:00)=(PST)|(-07:00)=(PDT)"/>
```

Arizona time, which doesn't have daylight savings:

```
<add key="timeZoneNames" value="(-07:00)=(MT)"/>
```

Step 3: Setup iService.MailPopper.config

The iService mailpopper service also requires a configuration file. The `iServicemailpopper.config` file is located in the For New Installation folder of the archive and must be copied into the folder that contains the Windows Service (same as the management console).

Edit the `iServicemailpopper.config` file to specify your connection details.

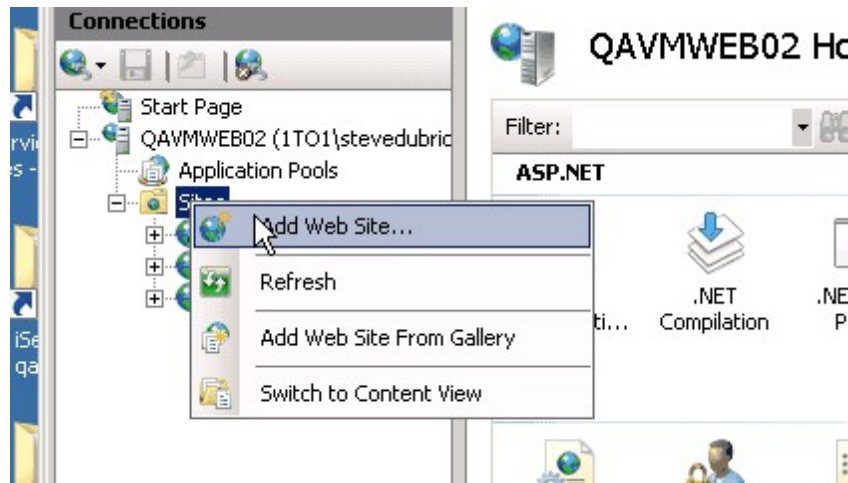
```
<databases>
  <database server="servername or IP" name="masterDBName" />
</databases>
```

The values should be the same as the web.config.

Step 4: Configure IIS

Open the **Internet Information Services (IIS) Manager**. This can be found in the Administrative tools folder usually located in the Control Panel or All Applications links in the Start Menu.

Once IIS is open, right click on the Sites folder and select **Add Web Site** ... from the popup menu.



Add website to IIS

Fill in the Site Name with the friendly name you'd like it to appear as in IIS. The physical path should match the location that you just placed the iService web site files (in our example, c:\inetpub\iService). Depending on your desired configuration the Bindings section can be modified to your requirements, but the most straightforward setup is to have a normal http connection on port 80 bind to a host name across all unassigned IP addresses for the server (this is shown in the example screenshot).

Add Web Site

Site name: iServiceAssistant Web Site Application pool: iServiceAssistant Web Site Select...

Content Directory

Physical path: C:\inetpub\iService ...

Pass-through authentication

Connect as... Test Settings...

Binding

Type: http IP address: All Unassigned Port: 80

Host name: iService.iServiceCRM.com

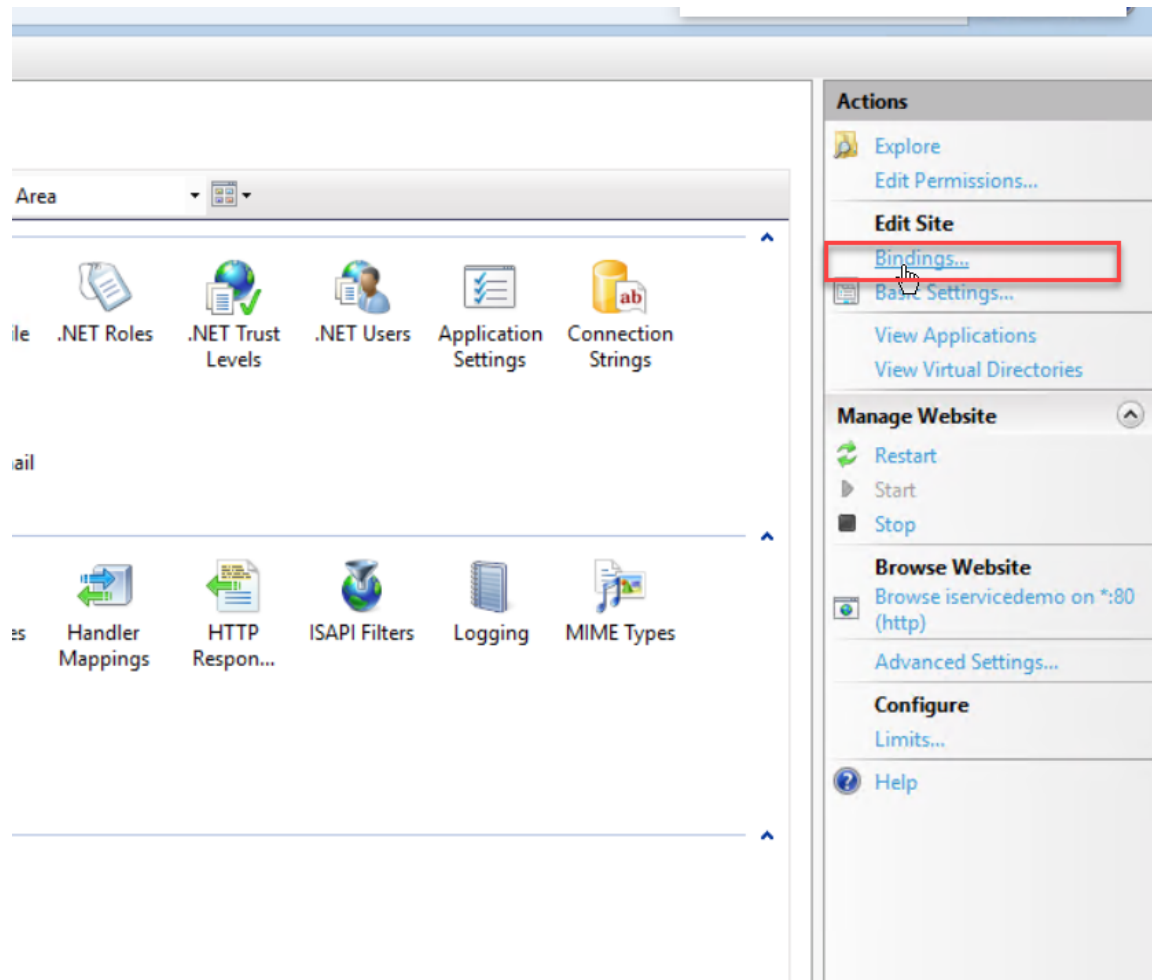
Example: www.contoso.com or marketing.contoso.com

☒ Start Web site immediately

OK Cancel

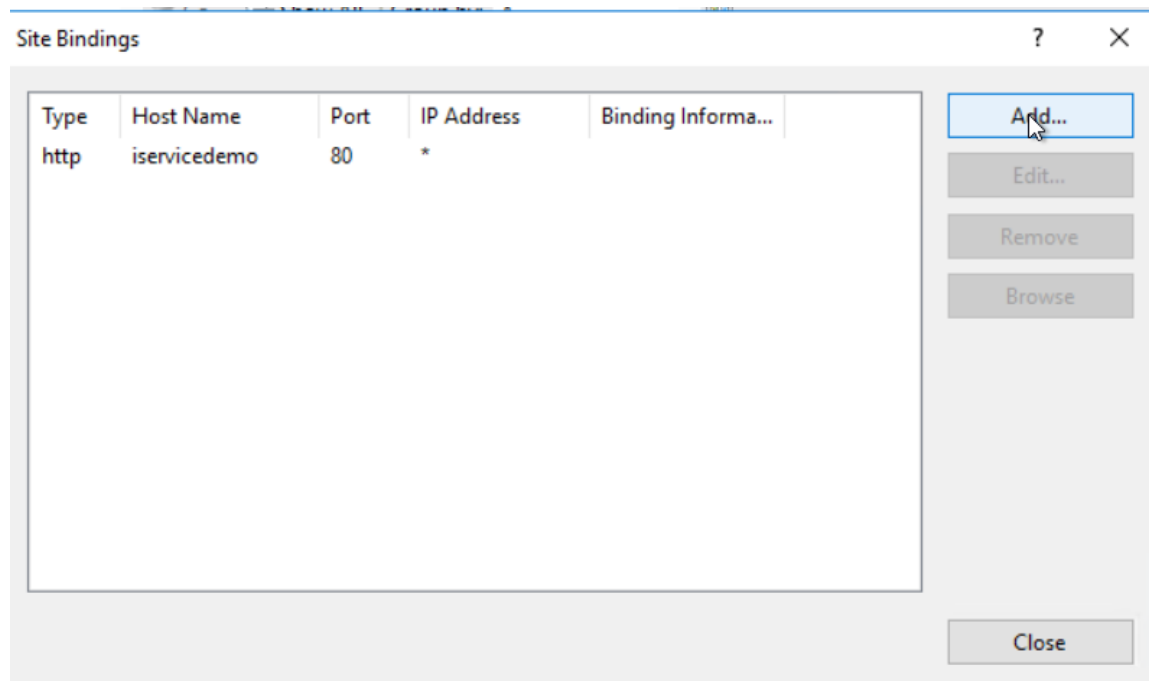
Website settings

The recommended configuration is to use HTTPS for all connections, and redirect any HTTP requests to use a secure connection. You will configure bindings for port 80 and 443. After the website is added, click on Bindings.



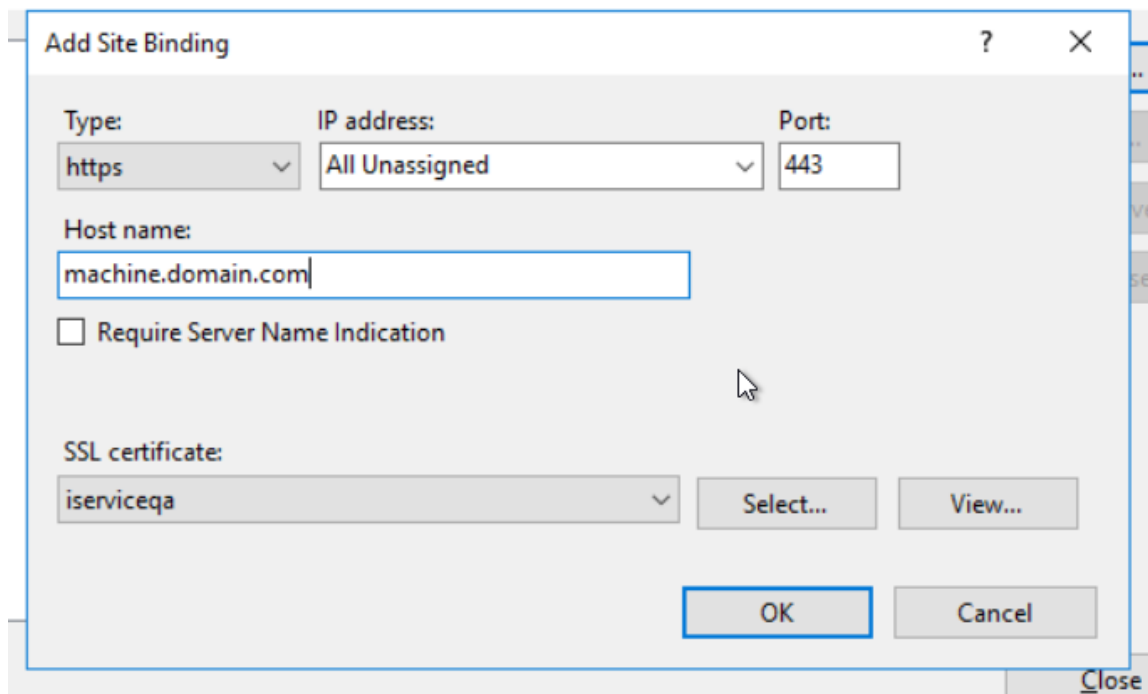
Open Bindings

Then click Add.



Add a binding

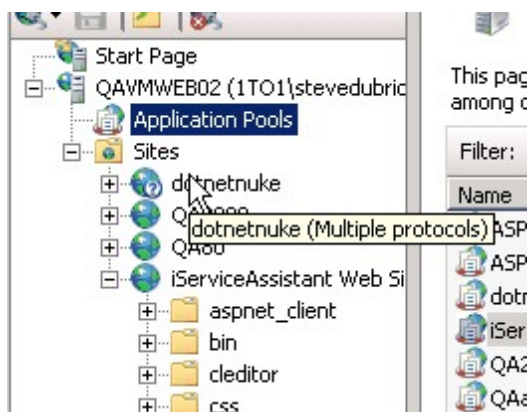
Configure port 443 by setting Type = https, Port = 443, and Host Name = your path to the website. Then select the SSL certificate from your server. You will need to import your SSL certificate to the server before it can be selected here.



Configure HTTPS

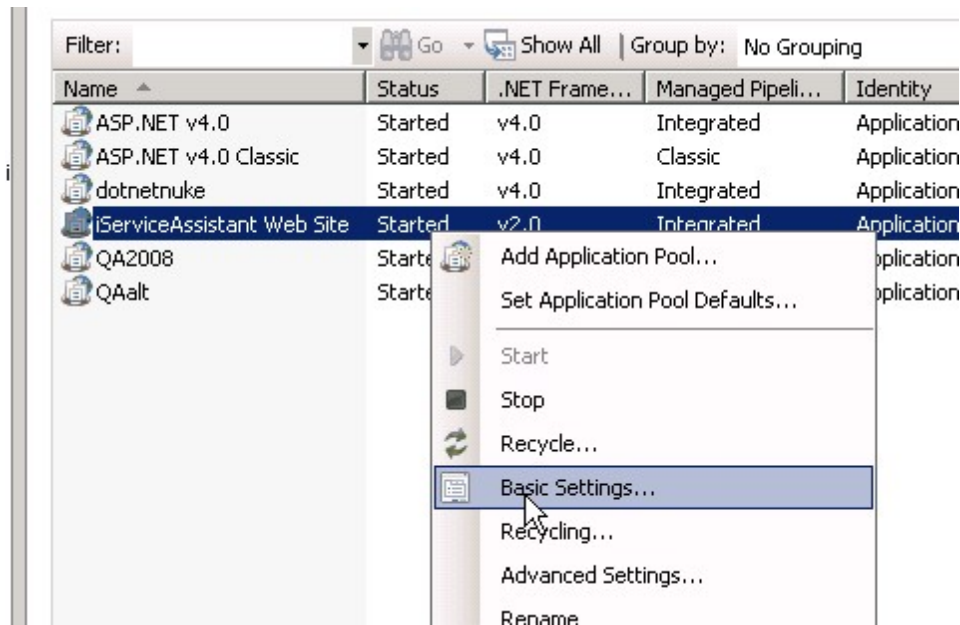
Click OK.

Next click on the **Application Pool** option on the left side of IIS:



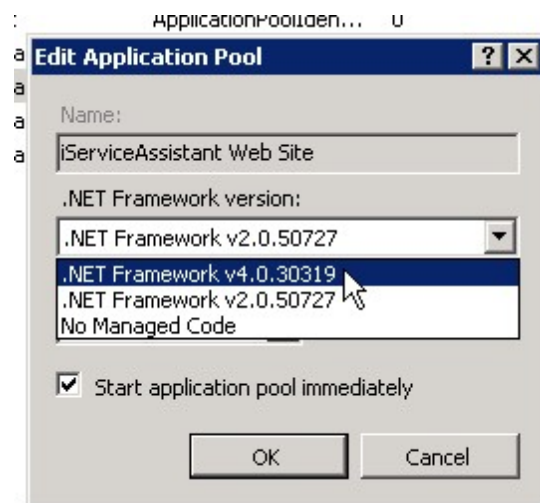
Application Pool

You should see a new application with the friendly name you gave the web site in the previous step. Right click this and choose **Basic Settings**.



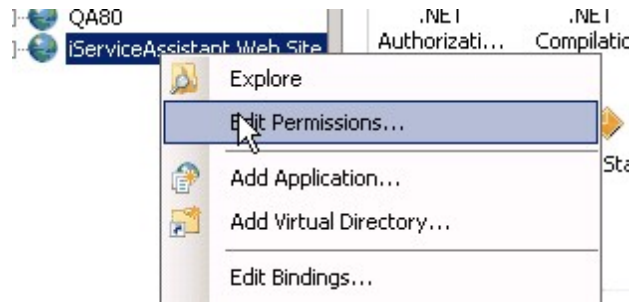
Basic Settings

iService uses .NET 4.x, so change the Framework version to the 4.x version, and click OK.



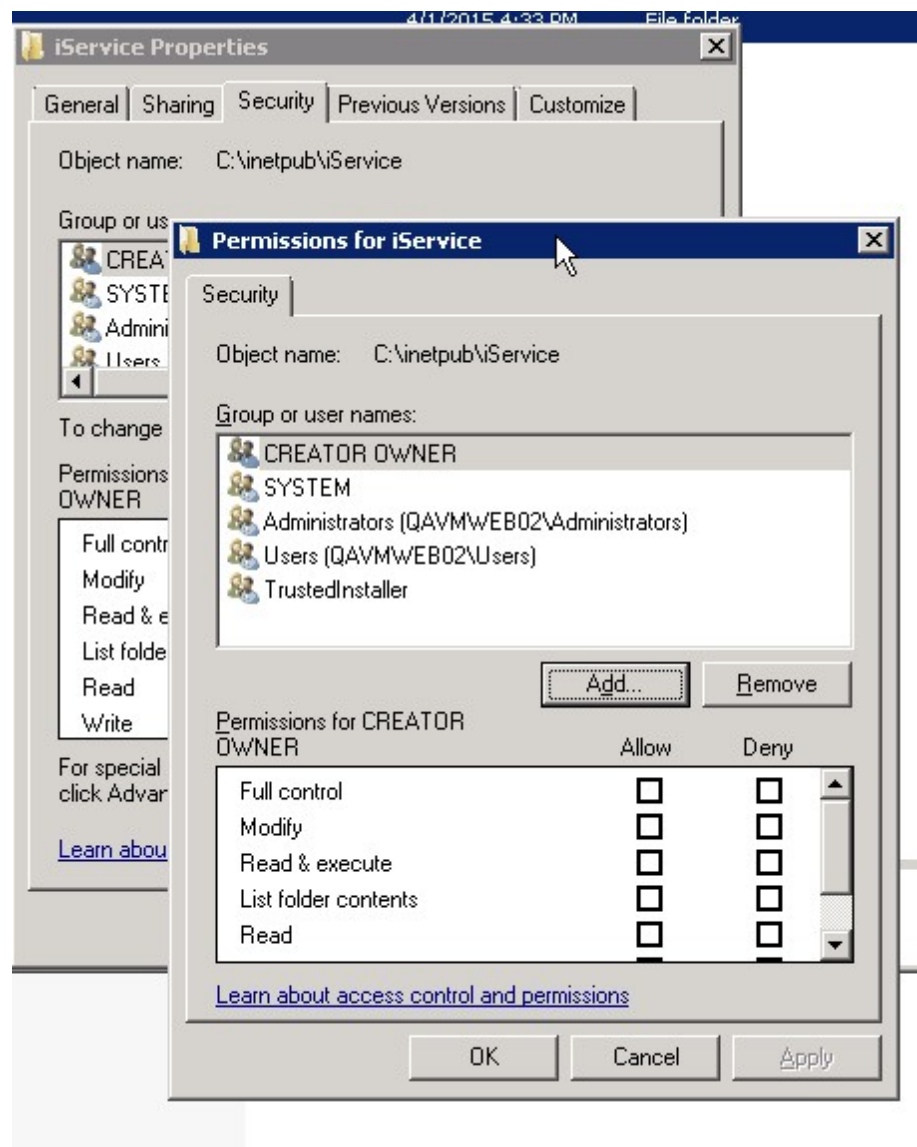
.NET Framework v4.x

Now we need to be sure that the new Application Pool has the proper permissions to access the folder you made in the earlier step. Right click on the web site in IIS and choose **Edit Permissions**.



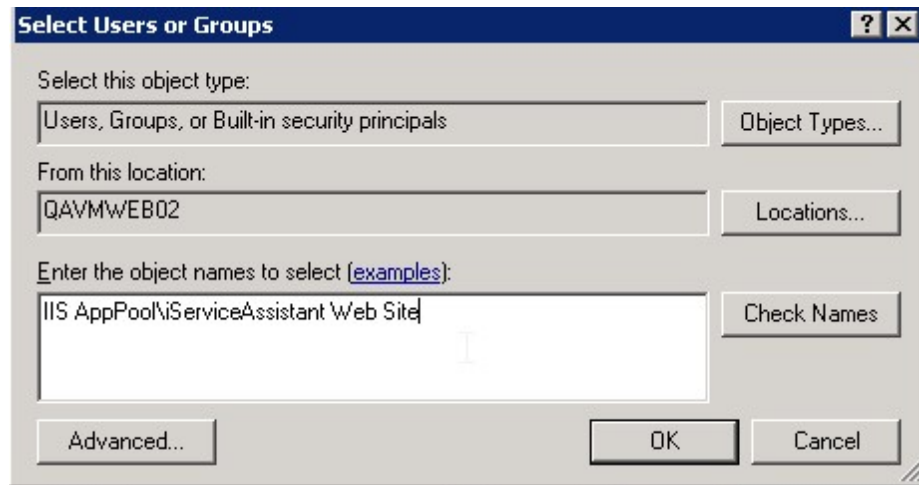
Edit Permissions

Choose the **Security** tab, and press the **Edit..** button. In the new window click **Add**.



Permissions for iService

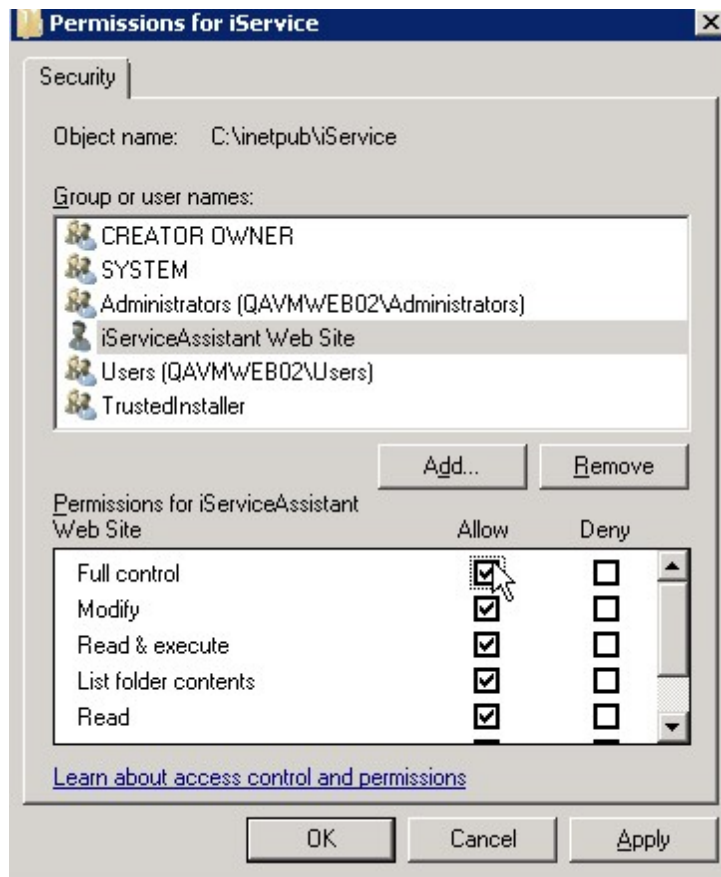
In the text area, type **IIS AppPool\<appname>** where <appname> is the friendly name you gave the website in the earlier step. In our example, it would be **IIS AppPool\iServiceAssistant Web Site**.



Select Users or Groups

Press OK.

Give the app pool **Full Control** permissions of the folder and Press OK through both windows.



Folder permissions for the iService website

2.1.7 Install Reports

iService includes a set of reports that provide important metrics for your business. These reports are written for Microsoft SQL Server Reporting Services. SSRS operates on its own server and uses a data source to connect to your iService tenant database. A separate set of reports can be configured with additional data sources to connect to multiple tenant databases.

A deploy utility is provided with the iService installation files to automate the deployment of reports. The utility can be used to create your data source, and will perform all of the required setup for your report environment. This section describes how to use the deploy utility for initial installation of reports.

Note: This process assumes that the release upgrade scripts (Deploy Tools) have been configured. You will perform the steps below on that server using the Release Archive from the installation or upgrade process.

Requirements and Prerequisites

The requirements for installing and running iService reports are described below. This is only required during initial installation.

Report Server - Microsoft SQL Server Reporting Services (SSRS) 2016 or higher is required. This requirement is for the version of SSRS only. The iService database is supported on all versions from SQL Server 2012 R2 or higher. For more information on configuring your SSRS server, refer to [Microsoft's SSRS documentation](#).

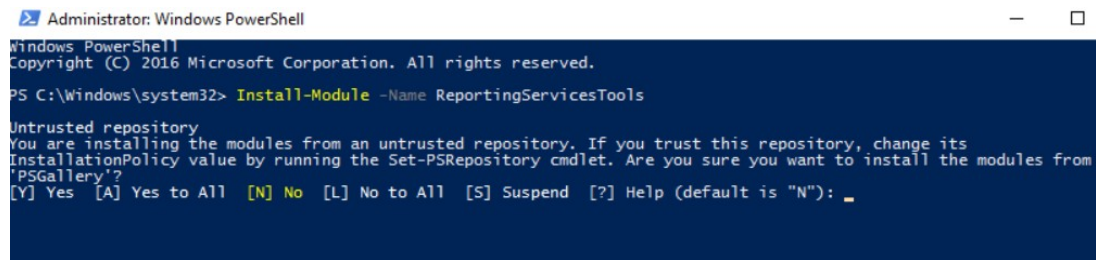
PowerShell - Microsoft PowerShell 3.0 or higher (included in any version of Windows Server 2012 and higher) is required to use the iService reports installation script.

SSRS PowerShell Tools Module - The installation scripts use this tool which is described in a GitHub repository at <https://github.com/microsoft/ReportingServicesTools>. The module is a PowerShell module provided by Microsoft that can be automatically configured using the command shown below.

Open a PowerShell prompt with Administrator privileges and run the following command:

```
Install-Module -Name ReportingServicesTools
```

You may be prompted once or twice to confirm the repository and nuget packages, press 'y' to continue.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Install-Module -Name ReportingServicesTools

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): _
```

Installing the SSRS Powershell Tools Module

The iService Reports files

iService reports are packaged with the iService Release Archive under the \Reports folder in the following structure:

\--Reports (folder)

| Deploy-
iServiceReports.ps1

The iService Reports PowerShell Deploy Utility

| iServiceCRM.sln

Visual Studio 2019 solution file for optional manual deploy

|
\---iServiceCRM
(Folder)

This folder contains the reports (.rdl) and other various resources used by the reports.

Reports Deploy Utility Command and Parameters

During the release upgrade, you copied the contents of the Release Archive to your web server. That archive contains the files you need to install or upgrade reports.

The reports utility (Deploy-iServiceReports.ps1) is a PowerShell script that accepts parameters to complete the various configuration tasks necessary to install reports. The basic format of the command is below.

```
./Deploy-iServiceReports.ps1
```

Utility Parameters

The iService Reports PowerShell Deploy Utility takes the following parameters. Other than the Data Source properties, all parameters are required, but if a parameter is not given will fall back to the specified default values. Parameters can be passed in using the following format:

```
ParameterName 'value'
```

The command accepts five parameters that are used for various configuration task. Parameters are entered using the format below.

```
./Deploy-iServiceReports.ps1 -parameterName 'parameterVal'
```

-ReportServerUri – Default value:

<http://localhost/ReportServer/>

This is the location of the SSRS that you are deploying to. If you are launching the deploy utility from the same server as SSRS is installed on, the default location can be used. Otherwise, a different server location can be specified.

-ReportFolderName – Default value: **iServiceCRM**

This is the folder that the reports will be deployed to on your SSRS server. This folder will be created if it doesn't already exist.

-DataSourcesFolderName – Default value: **Data Sources**

This is the folder that you store shared data sources in. All iService reports will reference the iService shared data source stored in this folder. This folder will be created if the -DeployDataSource parameter was specified and the folder doesn't already exist.

-DataSourceName – Default value: **iServiceCRM**

This is name of the iService shared data source that all deployed

reports will reference for connection details to the iService tenant database. Since a data source is required for all reports, the utility will halt deployment if it doesn't find a previously deployed data source with the specified name in the data sources folder AND the -DeployDataSource parameter wasn't given. A new data source with this name will be created or overwritten if the -DeployDataSource parameter was specified.

-DeployDataSource – (Optional) (No value necessary)

This parameter is a flag used to specify that you would like to either deploy a new iService data source or overwrite the existing iService data source. This is necessary for first-time deploys, or if you would like to change the connection details for an existing data source. ***If specified, the following parameters may optionally also be passed in. If they aren't passed in, the utility will prompt for input for the following:***

- **-DataSourceServerName** – (No default value)
The name of the SQL Server where your iService tenant database resides
- **-DataSourceDatabaseName** – (No default value)
The name of your iService tenant database
- **-DataSourceUserName** – (No default value)
A login on your SQL Server instance which has proper read access on the tenant database to run reports against.
- **-DataSourcePassword** – (No default value)
The password for the above login

Initial Installation

ProTip: Be sure your username you are running the utility has “Content Manager” permissions to the SSRS server you are deploying to. This permission allows you to write reports to the server. In general, any user in the Administrators group should automatically have this access by default.

Now we'll run the utility with the -DeployDataSource parameter. It will prompt us to enter the various connection details for this new data source to connect to the iService tenant database (in subsequent deploys, we can drop the -DeployDataSource parameter):

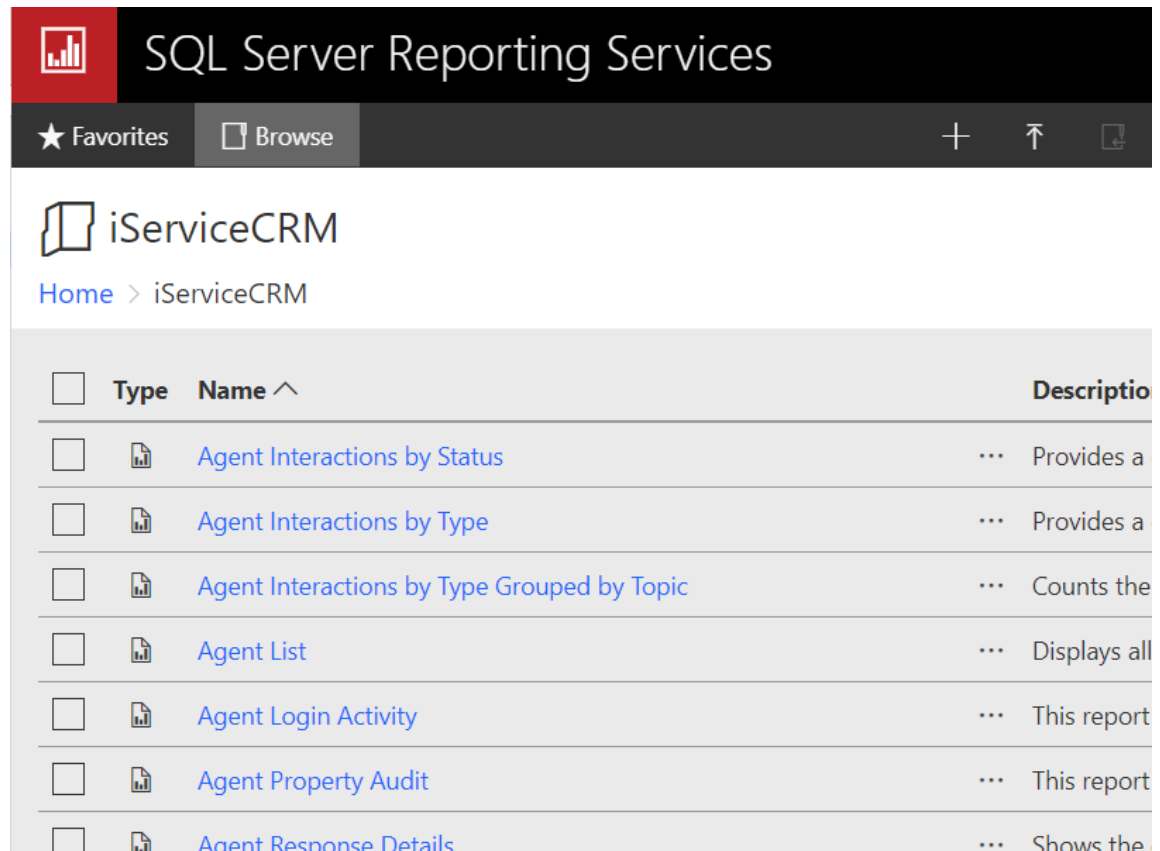
```
PS C:\iServiceReportsDeploy> .\Deploy-iServiceReports.ps1 -DeployDataSource  
  
cmdlet Deploy-iServiceReports.ps1 at command pipeline position 1  
Supply values for the following parameters:  
DataSourceServerName: sqlhost01  
DataSourceDatabaseName: iservice.tenantname  
DataSourceUserName: ssrs  
Data Source Password:: *****
```

Next, it will begin deploying. It will create the necessary folders and data source as needed, and upload the reports to the iServiceCRM folder on SSRS.

ProTip: If PowerShell throws an Execution Policy error, you may need to change your server's Execution Policy to RemoteSigned or Unrestricted using the Set-ExecutionPolicy command. For more details, see the [PowerShell documentation](#).

```
VERBOSE: Data source created successfully!  
An existing folder iServiceCRM was found on the report server.  
VERBOSE: Uploading C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Status.rdl to /iServiceCRM/Agent  
Interactions by Status  
VERBOSE: Performing the operation "Upload from C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Status.rdl to  
Report Server at /iServiceCRM" on target "Agent Interactions by Status".  
VERBOSE: C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Status.rdl was uploaded to /iServiceCRM  
successfully!  
VERBOSE: Uploading C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Type Grouped by Topic.rdl to  
/iServiceCRM/Agent Interactions by Type Grouped by Topic  
VERBOSE: Performing the operation "Upload from C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Type Grouped  
by Topic.rdl to Report Server at /iServiceCRM" on target "Agent Interactions by Type Grouped by Topic".  
VERBOSE: C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Type Grouped by Topic.rdl was uploaded to  
/iServiceCRM successfully!  
VERBOSE: Uploading C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Type.rdl to /iServiceCRM/Agent  
Interactions by Type  
VERBOSE: Performing the operation "Upload from C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Type.rdl to  
Report Server at /iServiceCRM" on target "Agent Interactions by Type".  
VERBOSE: C:\iServiceReportsDeploy\iServiceCRM\Agent Interactions by Type.rdl was uploaded to /iServiceCRM successfully!  
VERBOSE: Uploading C:\iServiceReportsDeploy\iServiceCRM\Agent List.rdl to /iServiceCRM/Agent List  
VERBOSE: Performing the operation "Upload from C:\iServiceReportsDeploy\iServiceCRM\Agent List.rdl to Report Server at  
/iServiceCRM" on target "Agent List".  
VERBOSE: C:\iServiceReportsDeploy\iServiceCRM\Agent List.rdl was uploaded to /iServiceCRM successfully!  
VERBOSE: Uploading C:\iServiceReportsDeploy\iServiceCRM\Agent Login Activity.rdl to /iServiceCRM/Agent Login Activity  
VERBOSE: Performing the operation "Upload from C:\iServiceReportsDeploy\iServiceCRM\Agent Login Activity.rdl to Report  
Server at /iServiceCRM" on target "Agent Login Activity".  
VERBOSE: C:\iServiceReportsDeploy\iServiceCRM\Agent Login Activity.rdl was uploaded to /iServiceCRM successfully!  
VERBOSE: Uploading C:\iServiceReportsDeploy\iServiceCRM\Agent Property Audit.rdl to /iServiceCRM/Agent Property Audit  
VERBOSE: Performing the operation "Upload from C:\iServiceReportsDeploy\iServiceCRM\Agent Property Audit.rdl to Report  
Server at /iServiceCRM" on target "Agent Property Audit".
```

After the utility is completed, you should be able to see the new reports on your SSRS server in the default folder /iServiceCRM



Other Installation Examples

```
./Deploy-iServiceReports.ps1
```

Deploys reports to the 'iServiceCRM' folder on the same server that the utility was run from. Expects to find an existing iService data source located on the server at /Data Sources/iServiceCRM which it will link all deployed reports to.

```
./Deploy-iServiceReports.ps1 -DeployDataSource
```

Deploys reports to the 'iServiceCRM' folder on the same server that the utility was run from. Will prompt the user for data source connection details, then create a new (or overwrite existing) data source at /Data Sources/iServiceCRM which it will link all deployed reports to.

```
./Deploy-iServiceReports.ps1 -ReportServerUri  
'https://www.MySSRSserver.com/ReportServer/' -  
ReportFolderName 'MyTenant' -DataSourcesFolderName 'DS' -  
DataSourceName 'MyTenant'
```

Deploys reports to the 'MyTenant' folder on an external SSRS server.
Expects to find an existing iService data source located
at /DS/MyTenant which it will link all deployed reports to.

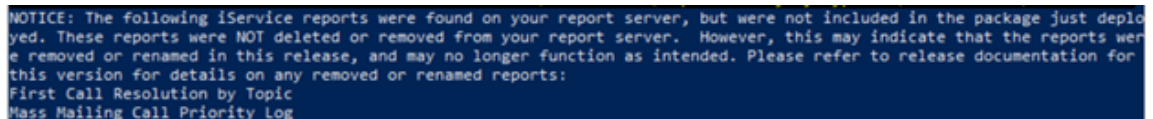
```
./Deploy-iServiceReports.ps1 -DeployDataSource -DataSource
```

Deploys reports to the 'iServiceCRM' folder on the same server that the
utility was run from. Will create a new (or overwrite existing)
datasource at /Data Sources/MyTenant that connects to the iService
tenant database named 'iService.myTenant' on the SQL Server instance
located at 'MyiServiceServer' with the SQL login 'ssrs'. It will then link
all deployed reports to this data source.

Deprecated and Removed Reports

Future report package releases may remove reports that were
contained in previous releases. This may be because multiple reports
were merged, reports were renamed, etc. The deploy utility will never
remove reports from your SSRS server, but will warn if it detects that
reports are no longer in the current release.

If reports were removed in the release deployed, you may see a notice
similar to the following:



```
NOTICE: The following iService reports were found on your report server, but were not included in the package just deployed. These reports were NOT deleted or removed from your report server. However, this may indicate that the reports were removed or renamed in this release, and may no longer function as intended. Please refer to release documentation for this version for details on any removed or renamed reports:  
First Call Resolution by Topic  
Mass Mailing Call Priority Log
```

Deprecated and removed reports

You can find more details on why the reports were removed in the [iService release notes](#). Generally, if they were removed from the release, they are no longer supported and should be deleted from the SSRS server.

2.2 Installing Upgrades (v8 and later)

Your iService Release Zip file includes an upgrade script for applying updates. After you configure the deploy tool, future updates will not require changes to the setup.

Always backup your iService databases before performing an upgrade.

Version Specific Upgrade Notes

Release 9.13

There are no release specific requirements when upgrading from 9.12 to 9.13.

Release 9.12

There are no release specific requirements when upgrading from 9.11 to 9.12.

Release 9.11

There are no release specific requirements when upgrading from 9.10 to 9.11.

Release 9.10

This release requires updates to the web.config file to support new libraries.

iService Version 9.10.0093 and higher require this change in web.config. Without this change, backend crashes might occur with an assembly mismatch warning in 3rd party integrations like OpenAI.

To fix, System.Text.Json needs to have assembly binding redirects changed from 8.0.0.0 to 8.0.0.4. Find these lines:

```
<assemblyIdentity name="System.Text.Json" publicKeyToken="7C954F7A2BD347F6" version="8.0.0.0" culture="neutral" />  
<bindingRedirect oldVersion="0.0.0.0-8.0.0.0" newVersion="8.0.0.4" />
```

change to:

```
<assemblyIdentity name="System.Text.Json" publicKeyToken="7C954F7A2BD347F6" version="8.0.0.0" culture="neutral" />  
<bindingRedirect oldVersion="0.0.0.0-8.0.0.4" newVersion="8.0.0.4" />
```

Release 9.9

There are no release specific requirements when upgrading from 9.8 to 9.9.

Release 9.8

There are no release specific requirements when upgrading from 9.7 to 9.8.

Release 9.7

There are no release specific requirements when upgrading from 9.6 to 9.7.

Release 9.6

This release requires updates to the web.config file to support new libraries.

Add the runtime section included below to web.config after the `</system.webServer>` section:

```
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-c
<dependentAssembly>
  <assemblyIdentity name="Newtonsoft.Json" publi
    <bindingRedirect oldVersion="0.0.0.0-7.0.0.0"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Runtime" publi
    <bindingRedirect oldVersion="0.0.0.0-4.1.1.1"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Reflection" pu
    <bindingRedirect oldVersion="0.0.0.0-4.1.1.0"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Net.Sockets" p
    <bindingRedirect oldVersion="0.0.0.0-4.1.1.0"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Security.Crypt
    <bindingRedirect oldVersion="0.0.0.0-4.1.0.0"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Net.Http" publi
    <bindingRedirect oldVersion="0.0.0.0-4.1.1.3"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Security.Crypt
    <bindingRedirect oldVersion="0.0.0.0-4.0.1.0"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Security.Crypt
    <bindingRedirect oldVersion="0.0.0.0-4.1.1.2"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Runtime.Compil
    <bindingRedirect oldVersion="0.0.0.0-6.0.0.0"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Text.Json" pub
    <bindingRedirect oldVersion="0.0.0.0-8.0.0.0"
  </dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Text.Encodings
    <bindingRedirect oldVersion="0.0.0.0-8.0.0.0"
```

```

</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Xml.XmlDocumen
    <bindingRedirect oldVersion="0.0.0.0-4.0.2.0"
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Reflection.Type
    <bindingRedirect oldVersion="0.0.0.0-4.1.5.0"
</dependentAssembly>
</assemblyBinding>
</runtime>

```

Release 9.5

When updating to v9.5, you must ensure your database is running in the proper compatibility mode.

iService 9.5 uses [OPENJSON\(\)](#) to parse JSON bodies stored in interaction bodies, which requires the tenant database to have a compatibility level of 130 (sql 2016) or higher before the tenant can upgrade. The upgrade will fail otherwise.

For basic steps on how how to set compatibility level for database see <https://learn.microsoft.com/en-us/sql/relational-databases/databases/view-or-change-the-compatibility-level-of-a-database?view=sql-server-2017>

For more in depth instructions, including potential impacts, see <https://learn.microsoft.com/en-us/sql/t-sql/statements/alter-database-transact-sql-compatibility-level?view=sql-server-2017>

Release 9.4

There are no release specific requirements when upgrading from 9.3 to 9.4.

Release 9.3

There are no release specific requirements when upgrading from 9.2 to 9.3.

Release 9.2

There are no release specific requirements when upgrading from 9.1 to 9.2.

Release 9.1

This release includes a replacement for the framework used for real-time browser updates. Prior versions used WebSync for that process but version 9.1 includes a new custom web socket system. This requires new lines to be added to your web.config file in the iService Website. The only changes to web.config are the addition of four new lines highlighted in green (nothing needs to be removed).

```
@@ -11,6 +11,8 @@
    <system.web>
        <httpHandlers>
            <add verb="*" path="websync.ashx" type="FM.
            <add verb="*" path="web.skt" type="OneToOne
            <add verb="*" path="longpoll.skt" type="One
<!-- CORS Support. Uncomment this block and the other blo
            <add verb="*" path="*.api" type="
-->
@@ -26,6 +28,8 @@
    <validation validateIntegratedModeConfiguration=
    <handlers>
        <add verb="*" path="websync.ashx" type="FM.
        <add verb="*" path="web.skt" type="OneToOne
        <add verb="*" path="longpoll.skt" type="One
<!-- CORS Support. Uncomment this block and the other blo
        <add verb="*" path="*.api" type="OneToOne.i
-->
```

Release 9.0

There are no release specific requirements when upgrading from 8.8 to 9.0.

Release 8.8

The .NET Framework versions 4.5.2, 4.6, and 4.6.1 will reach end of support on April 26, 2022. After this date, security fixes, updates, and technical support for these versions will no longer be provided. Beginning with release 8.8, the web.config file for your iService website should be updated to use .Net 4.6.2.

Look for this line in your `web.config` file and update to 4.6.2. However, you must first ensure your server is running the 4.6.2 version of .Net.

Change the old value:

```
<httpRuntime maxRequestLength="16384" requestValidationMo
```

to the new value:

```
<httpRuntime maxRequestLength="16384" requestValidationMo
```

Releases 8.3 through 8.7

There are no release specific requirements when upgrading to 8.3 through 8.7.

Release 8.2

Release 8.2 includes initial support for retrieving email using Microsoft Exchange Web Services (EWS), often called Exchange Online or hosted Exchange. This release contains a defect that removed a constraint preventing the creation of duplicate mailbox configurations in a tenant. This defect is corrected in 8.3, so when upgrading beyond 8.1 you should skip the 8.2 installer and go directly to 8.3.

If you previously installed 8.2 and are missing the constraint, you can add it back using the script below.

```
ALTER TABLE [dbo].[Mailboxes] ADD CONSTRAINT [U_Mailboxe  
(
```

```
[inServer] ASC,  
[inUsername] ASC  
)
```

Release 8.1

The web.config file located in your website installation folder should be modified to enable modern encryption during mail processing. Without this change, iService may only process mail using older, non-secure encryption protocols.

The following line should be changed from:

```
<httpRuntime maxRequestLength="16384"  
requestValidationMode="2.0" />
```

Changed to:

```
<httpRuntime maxRequestLength="16384"  
requestValidationMode="2.0" targetFramework="4.6.1"/>
```

Release 8.0

Update to the web.config file for your iService website

A location Service Worker text block needs to be added inside the `<configuration>` tag in the web.config file. This can be placed between `</system.webServer>` and `</configuration>` tags at the end of the file:

```
</system.webServer>
```

insert the section below

```
  <location path="js/serviceWorker.js">  
    <system.webServer>  
      <httpProtocol>  
        <customHeaders>  
          <add name="Service-Worker-Allowed" value="/" />  
        </customHeaders>  
      </httpProtocol>  
    </system.webServer>  
  </location>
```

```
</configuration>
```

Check for duplicate property values

In certain conditions, the auto save process may create duplicate values for properties that are set to a single value. The following constraints were added in 8.0 to prevent this at the database level.

Case properties:

```
ALTER TABLE [InteractionsCasesProperties] ADD CONSTRAINT
```

Interaction Properties:

```
ALTER TABLE [InteractionsProperties] ADD CONSTRAINT U_Int
```

You can determine if your database already has this constraint using the script below. These will return a single NULL if the constraint doesn't exist, or a single id if it does:

Case properties:

```
SELECT OBJECT_ID('dbo.[U_InteractionsCasesProperties_thre
```

Interaction properties:

```
SELECT OBJECT_ID('dbo.[U_InteractionsProperties_interacti
```

If your v7.8 database contains interactions with this condition (multiple values for a property that is not set as multiple value), you must correct them before upgrading to v8. You can locate interactions with this issue using the scripts below.

To check duplicate case property values:

```
SELECT  propertyid,
        threadid,propertyvalue,valueindex,count(threadid)  '
FROM    [InteractionsCasesProperties]
GROUP BY propertyid,
        threadid,propertyvalue,valueindex
HAVING  count(threadid) > 1
ORDER BY count(threadid) desc, threadid desc
```

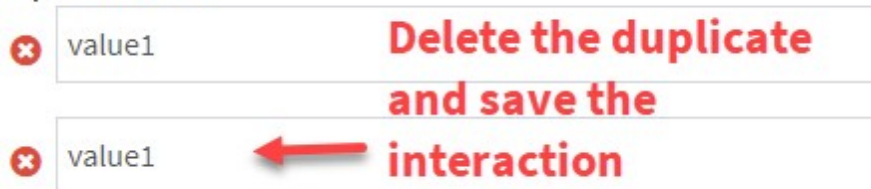
To check duplicate interaction property values:

```
SELECT  propertyid,
        interactionid,propertyvalue,valueindex,count(intera
```

```
FROM [InteractionsProperties]
group by propertyid,
        interactionid,propertyvalue,valueindex
having count(interactionid) > 1
order by count(interactionid) desc, interactionid desc
```

The quickest way to correct this issue once you locate the IDs (using the script above) is to remove the duplicates using the Search Interactions page.

Important Data



Removing a duplicate property

Configuring the Deploy Tool






Create the Deploy Tool Folders

The deploy tool must be run on the Windows server that is running the iService web applications. Create a folder named \deployiservice on your iService web server. Then, unzip the Release Archive and copy the contents of the Deploy Tools folder into \deployiservice.

Name	Type
Binaries	File folder
Deploy Tools	File folder
For New Installation	File folder
iService Utilities	File folder
Reports	File folder
Website	File folder
iServiceSetup.exe	Application

The Release Archive

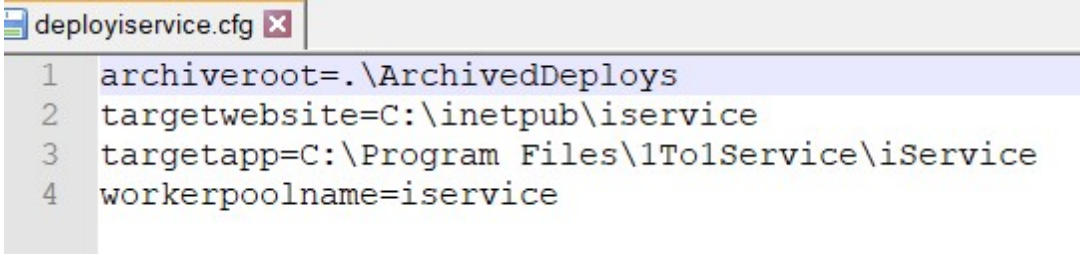
The contents of the Deploy Tools folder should be as shown below.

Name	Type
 ArchivedDeploys	File folder
 iservicesetupfiles	File folder
 deployiservice.bat	Windows Batch File
 deployiservice.cfg	CFG File
 wtee.exe	Application

Contents of the Deploy Tools folder

Edit the deployiservice.cfg file

The deployiservice.cfg file includes four parameters as shown below.



```
1 archiveroot=.\ArchivedDeploys
2 targetwebsite=C:\inetpub\iservice
3 targetapp=C:\Program Files\1To1Service\iService
4 workerpoolname=iservice
```

The configuration file

When the upgrade is run, the prior installation (website, mail popper files, and the prior setup files used during upgrade) is backed up to an ArchivedDeploys folder. The first line refers to the location where these files will be saved. Do not modify this line.

The second line references the location of the iService website. This is typically within the \inetpub folder, but may be configured differently for your installation.

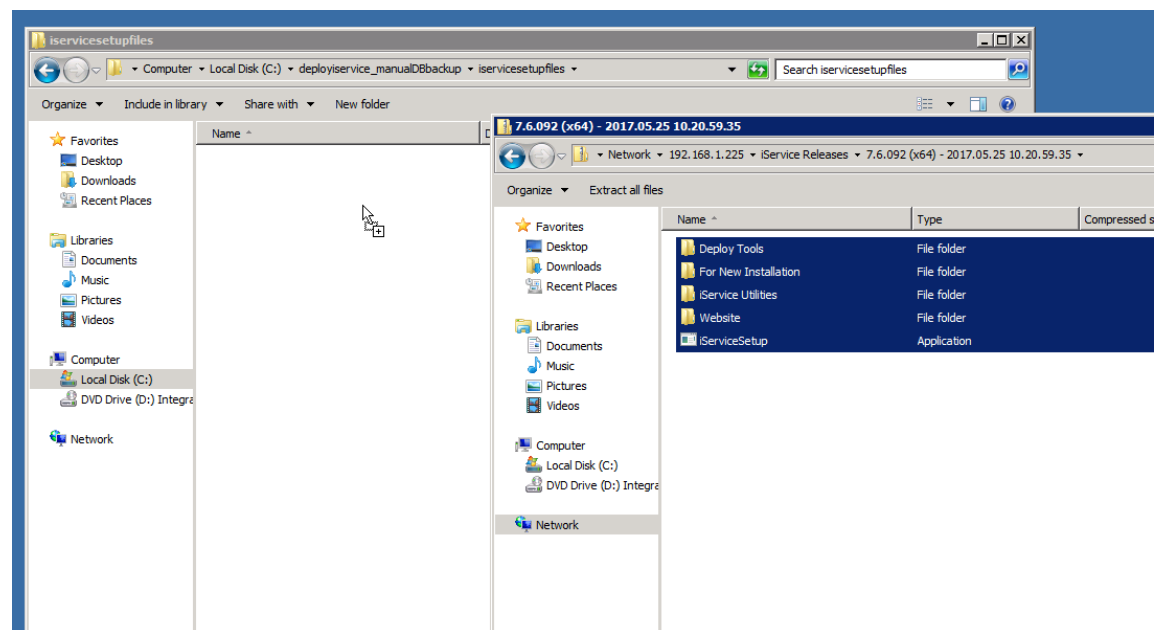
The third line references the location for the mailpopper files. These should be located at the path shown here and should not be changed.

The fourth line is the name of the IIS worker pool for iService. The name of your worker pool was established during the initial installation. This value must match the name of your worker pool.

Running the Deploy Tool

Extract the Release zip contents

Open and copy the entire contents of the release .zip file to the "iservicesetupfiles" sub-folder created during configuration of the tool. Generally this is something like c:\deployiservice\iservicesetupfiles.

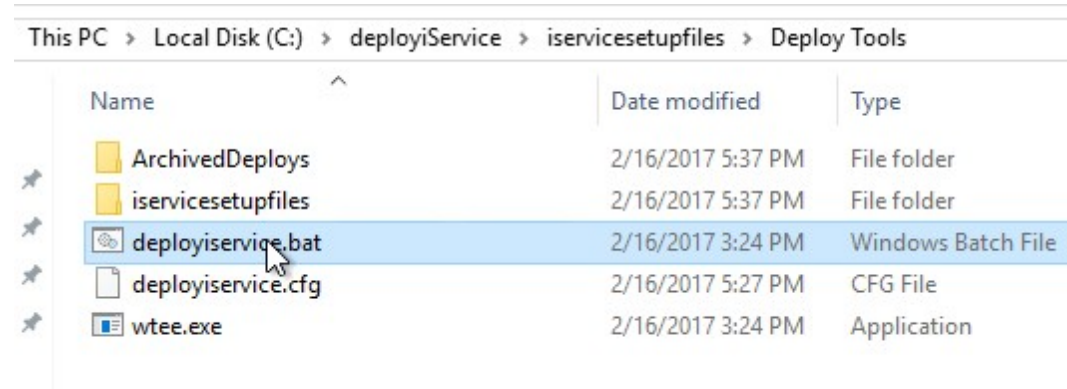


Click to expand

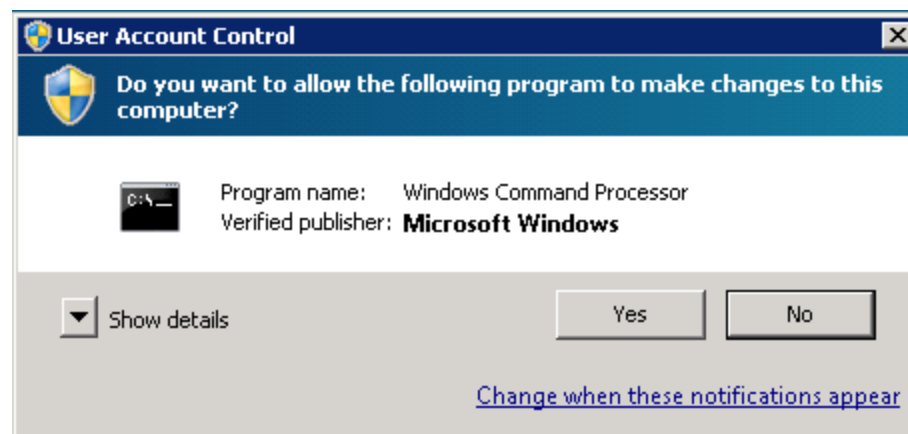
Run the deployiservice.bat file

The batch file for running the upgrade should have been copied to the \deployiservice folder during your configuration. It is also located in the release .zip as shown below.

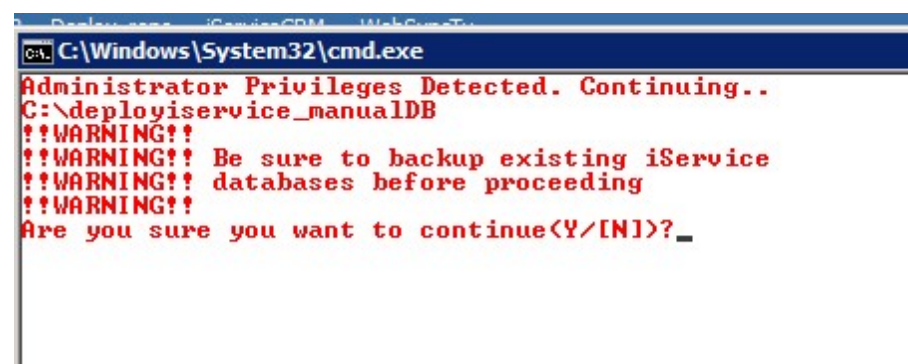
*Note that this step is not necessary for subsequent upgrades.



Right click and run the new deployiservice.bat file as Administrator.

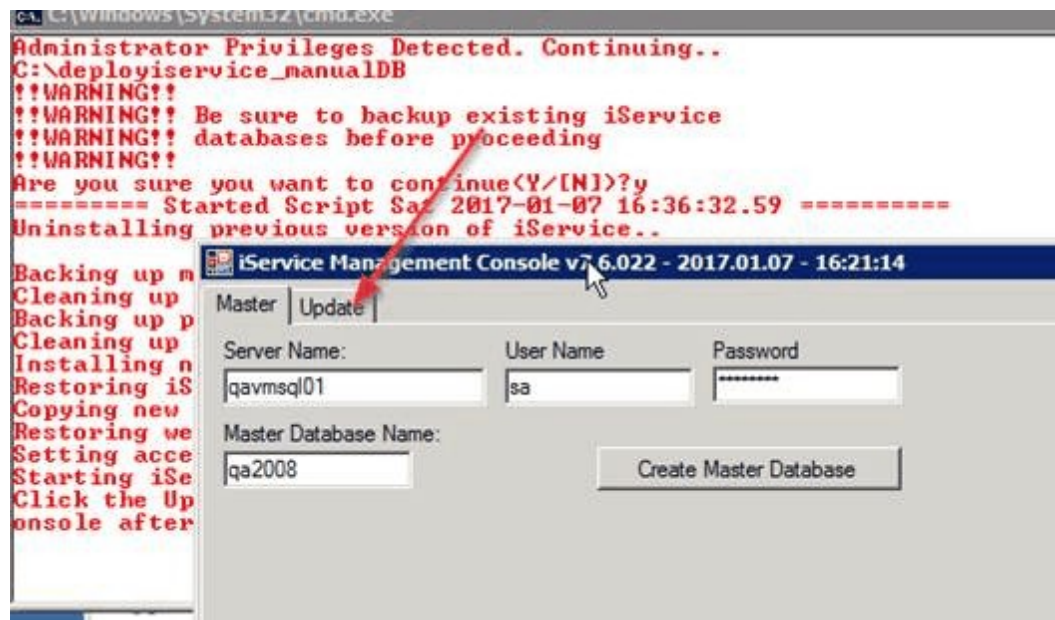


As a safety precaution. You will be reminded to backup your databases before continuing. Press Y and enter to continue.

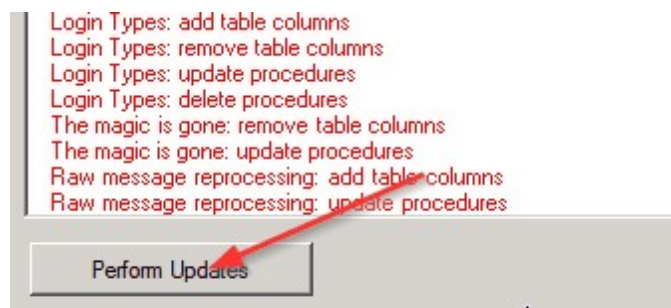


Upgrade database via the management console

During the upgrade, the management console will appear. Click on the "Update" tab to process any required database updates.



If any database updates are needed, they will be listed in red text. Click the "Perform Updates" button to apply them:



After they are applied, close the Management Console.

Complete upgrade

When the deploy script says "Done. Press any key to continue..", the upgrade is finished and you can close the deploy script window.

```
Done. Log File and old deployment archived to .\ArchivedDeploys\2017.01.07 16.36
.32.59\
Press any key to continue . . .
```

2.3 Upgrading Reports

Reports are upgraded using the same files and utilities as the initial installation. The parameters required depend on your local environment, but will generally be the same for upgrades as they were for your initial installation.

[Read about the installation scripts used for upgrading.](#)



Technical Configuration and Tools

3 Technical Configuration and Tools

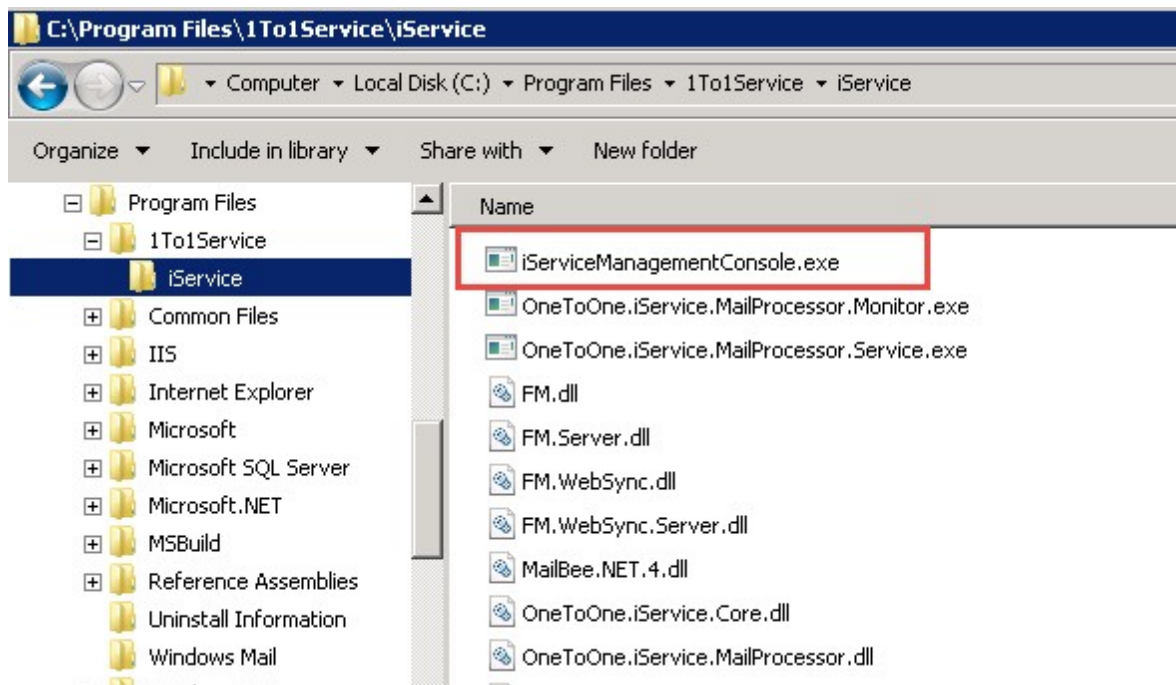
Although the majority of configuration is done through the iService website, there are a few tools that require network administrator access. The [Management Console](#)⁶⁶ provides various features for changing the superuser password, posting broadcast messages to tenants, updating agent counts, etc. The [Mail Popper](#)⁸¹ monitor doesn't have any configuration options, but provides a real-time view of the iService threads that are running.

There are various integrations available to third party products. These integrations require your organization to create the configuration in the third party application and then enter the required authentication keys into the [External APIs](#) page of your tenant.

3.1 Management Console

The Management Console is used to create new tenants, apply upgrades to the system, change certain configuration settings for tenants, and includes a broadcast message feature for sending notices to users logged into tenants. It is a Windows application that can be run from any Windows machine that can connect to the iService database server.

By default, its executable is located in the same directory as the iService Mail Popper service and iService Mail Monitor (the default location is C:\Program Files\1To1Service\iService).



The iService Management Console app

Running the Management Console from the command line (for automation)

The application supports the following optional parameters.

```
iServiceManagementConsole [-nogui "server_name" "master_d
```

Use the -nogui parameter and the login details to run the management console from the command line. Once launched in the -nogui mode, the application will connect to your master data and run any database updates defined for its version. The -nogui command line version is designed to be used for automated scripts, or when applying version updates that don't require tenant changes. The full GUI version is required to make changes, such as updating license counts or broadcast messages.

The program will return an exit code upon completion. When the application connects and performs the database updates successfully,

a return code of 0 is provided. If an error occurs during execution, a return code of 1 is provided. To determine the nature of the error, run the management console with the full GUI to observe the errors.

Using the Graphical User Interface

The management console GUI is used for installing and upgrading iService, updating license counts, importing tenants, and sending broadcast messages to users.

Master Tab

The Management Console is used to create the initial iService database and update the various databases and stored procedures for new releases. The iService master database contains reference information for all tenant databases. The Master tab includes the following fields, and requires a SQL Server login with DB Owner privileges to the database. This level of access is required because the release updates must have the rights to modify the database.

Enter your installation details and then click the "Update" tab to continue. The Update tab will verify that all of your databases have been updated to this build version.

The Master Tab in the management console

Server Name - The location of the database server that will contain the iService master database and all of its tenant databases.

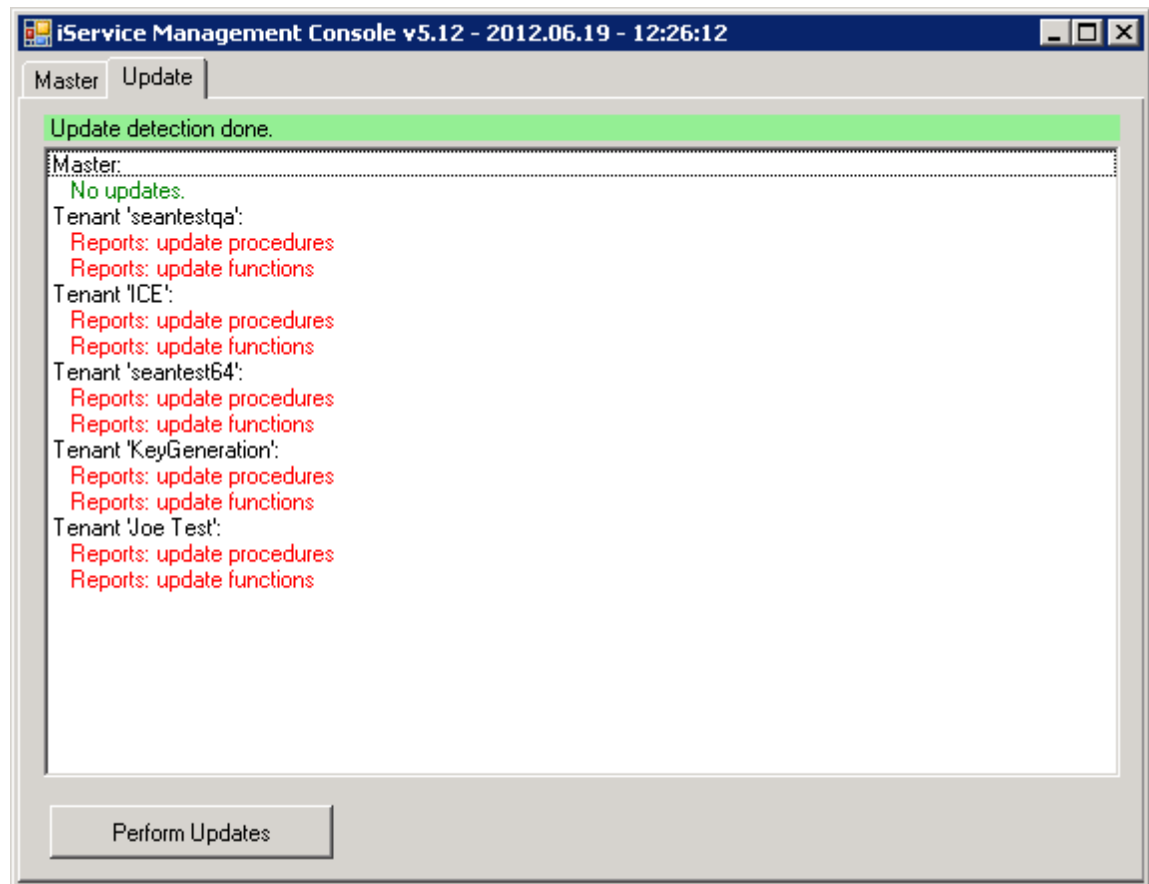
User Name / Password - The user name and password within SQL Server required to login to the iService database.

Master Database Name - The name that will be used for the iService master database.

Create Master Database - This button is used to create the initial master database and can also recreate a new version for testing purposes. **If you click this button after the initial installation, it will delete your database and should only be used during initial installation.**

The Update Tab

The Update tab is used to apply database updates to the iService master and tenant databases. Database updates are delivered in the form of an update to the Management Console, and clicking the Perform Updates button will apply the necessary updates to the Master and all tenant database. The tab is required every time the console loads to ensure databases are updated.



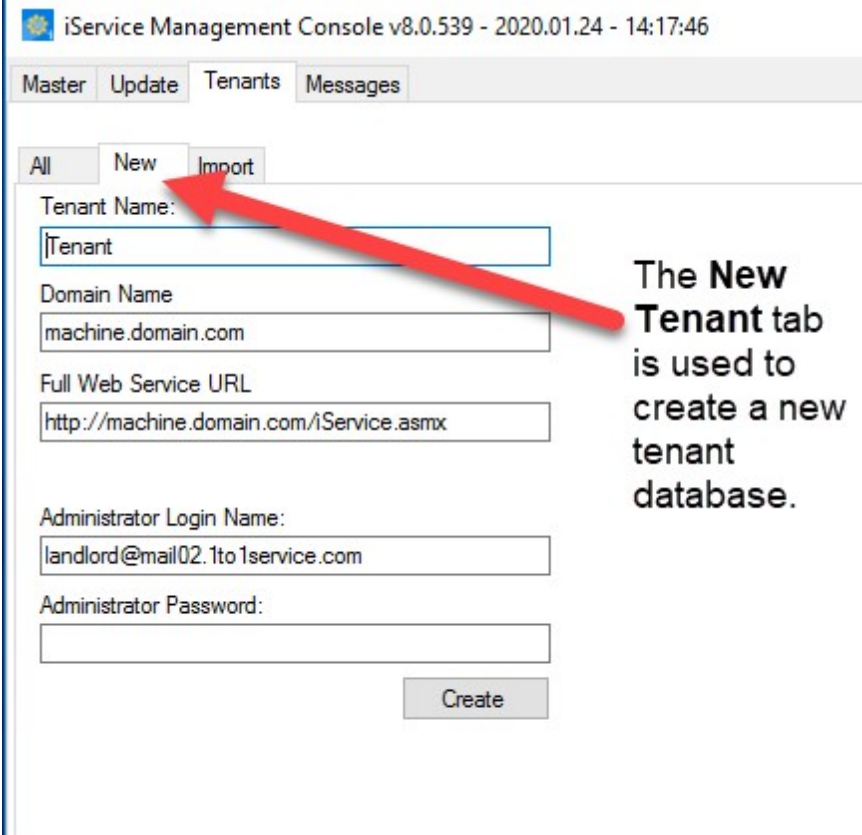
The Update tab in the Management Console

The Tenants Tab

The tenants tab provides a list of tenants that are part of this installation. It includes buttons to create new tenants, import tenants from another master, delete tenants, and modify existing tenants.

Creating New Tenants

To create a new tenant, select the New tab and enter the details for the new tenant.



iService Management Console v8.0.539 - 2020.01.24 - 14:17:46

Master Update **Tenants** Messages

All **New** Import

Tenant Name: [Tenant]

Domain Name [machine.domain.com]

Full Web Service URL [http://machine.domain.com/iService.asmx]

Administrator Login Name: [landlord@mail02.1to1service.com]

Administrator Password: []

Create

The New Tenant tab is used to create a new tenant database.

The New Tenant tab in the management console

Tenant Name – This must be a unique name, and will be used for the tenant database name. Enter only the tenant name here. The name of the master database will be prepended to form the full database name (the naming convention for tenant databases is

`masterDBname.tenantDBname`).

Domain Name – The Domain Name specifies the domain where users of the iService system will login and must be unique for each tenant.

Full Web Service URL – This is the location of the web services file for the new tenant. It should typically be the domain name with http or https (as appropriate) followed by /iService.asmx (the name of the web services file).

Administrator Login Name – Enter a login name for the initial superuser of the iService website. This will be contactID = 1 within the new tenant.

Administrator Password – Enter a password for your initial superuser.

Click the Create button to create your tenant database. Once the database is created and your DNS is setup, you should be able to load the iService website and login using this administrator user ID.

Importing a Tenant

The Import Tab allows the importing of a previously set up tenant. This can be used to move tenants from other servers.

The first step in importing a tenant is to restore the backup of the iService tenant database onto the new SQL Server. Ensure the proper naming convention is used to match the naming of the master database for the server on which you are importing the tenant. The naming convention is MasterDBName.TenantName.

Click on the Tenants tab and then click “Import Tenant.” This will open the dialogue box shown below.

Importing a tenant

Enter the name of the database in the Name of Tenant Database box (without the master database prefix) , and click the Next button. For example, if the tenant database name is iservice.MyTenant, you would enter only MyTenant. When you click Next, the Desktop App will attempt to connect to the database. If it succeeds in connecting to the database, it will verify whether database updates are available.

1. Enter the name of the database in the Name of Tenant Database box (without the master database prefix) , and click the Next button. For example, if the tenant database name is iservice.MyTenant, you would enter only MyTenant. When you click Next, the Desktop App will attempt to connect to the database. If it succeeds in connecting to the database, it will verify whether database updates are available.
2. If the tenant needs to be updated, the “Update” heading should show the option to upgrade it. Click the Perform Updates button and

iService will apply the necessary updates to bring the tenant up to the version of the current installation.

NOTE: You can only roll a tenant database forward to a newer version using the Perform Updates button. You cannot roll a database backwards to a prior version of iService.

3. “Website Domain Conflicts” allows the tenant to have its prior associated domain changed to eliminate conflicts. If a conflict is found, you can type a new domain name for the tenant and then click Next.
4. Once the updates are applied and the domain is correct, click the Add Tenant to Master button to complete the import process.

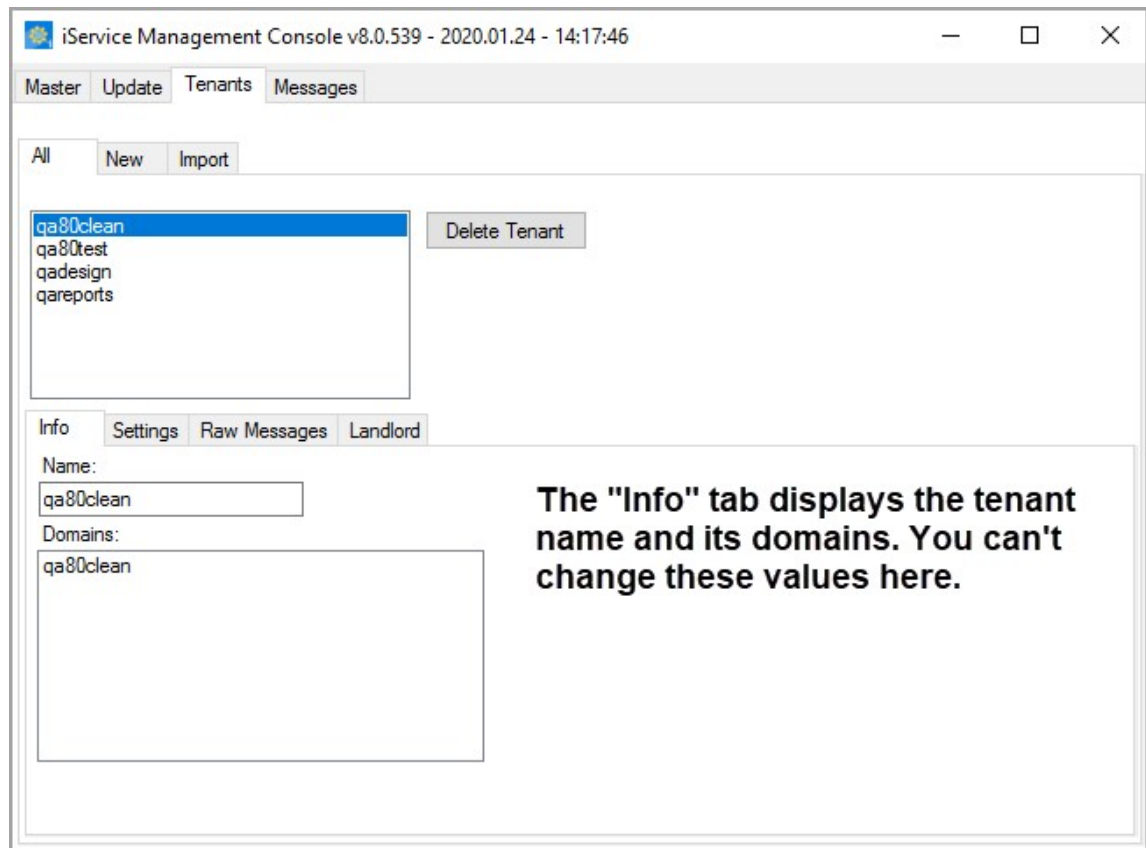
“Import Tenant” will attach the desired tenant to the Master database.

Modifying a Tenant

The Tenants tab provides a list of existing tenants. Select a tenant to view or modify its settings.

Info Tab

The tenant info tab displays the database name and all of the domains associated with the tenant's websites. These domains are set within each tenant in the Admin Tools – Websites tab.

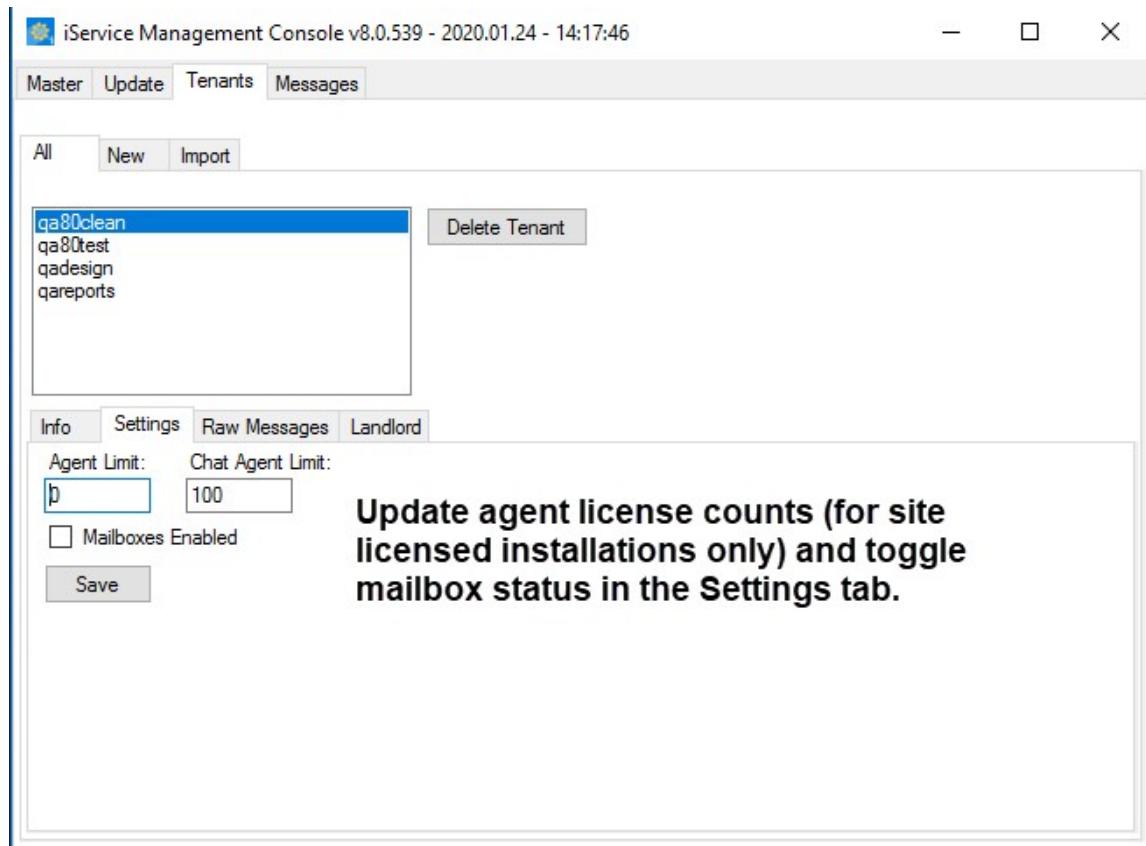


The Info Tab

Settings Tab

The settings tab displays the number of agents licensed to use the tenant within the Agent Limit and Chat Agent Limit text boxes. The number of agents can only be modified if you have an unlimited user site license.

A checkbox is provided to enable mailboxes. You can disable retrieval for all mailboxes within a tenant regardless of the status set for each mailbox within the Admin Tools - Segment - Mailboxes tab. This should be checked to enable mailboxes in most situations.



The Settings tab in the Management Console

Raw Messages Tab

NOTE: When extracting raw messages you should first create an empty folder into which you will extract the messages.

Extract Raw Messages Button

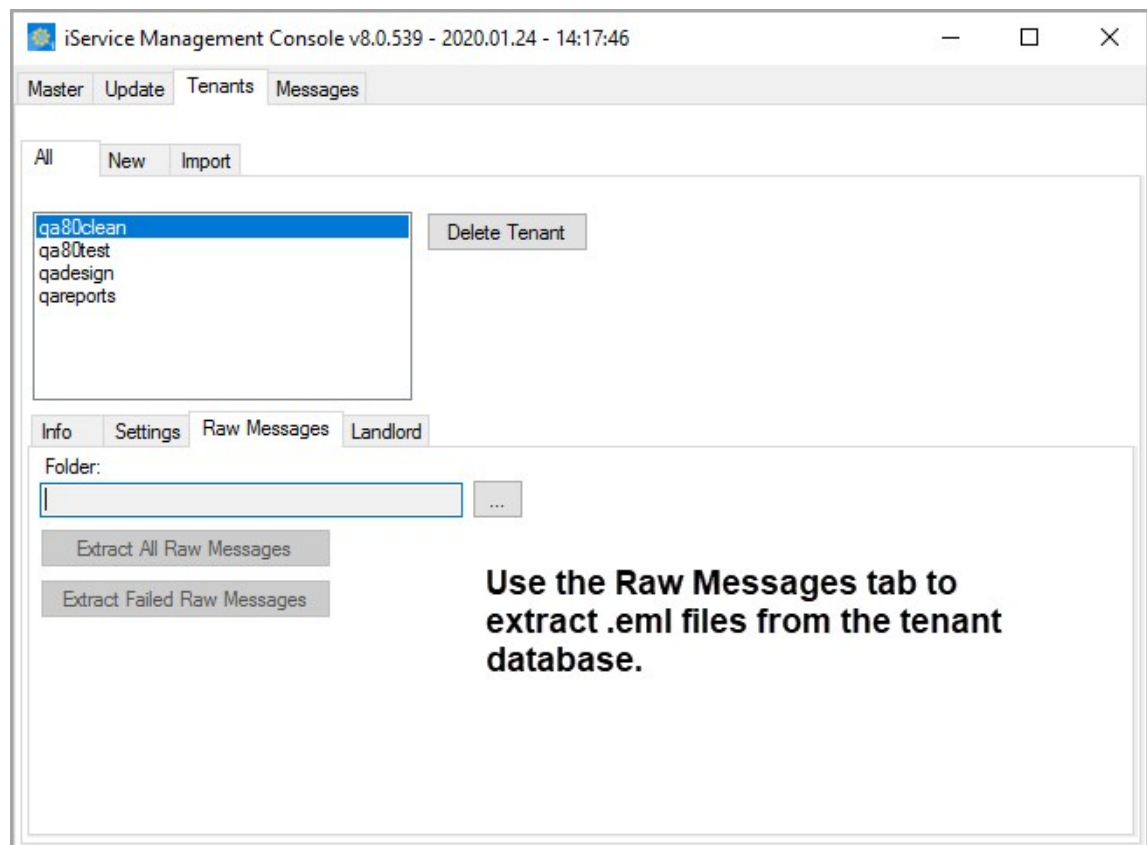
The Extract Raw Messages tab button is used to extract all of the raw e-mail messages from a tenant database into a folder. Specify the directory into which the messages should be extracted and click the Extract Raw Messages button to begin the extraction process. Messages will be extracted in .eml format as a separate file for each email message.

This process will extract ALL raw messages from your tenant. Use with caution as the result could be an

extremely large number of files.

Extract Failed Raw Messages Button

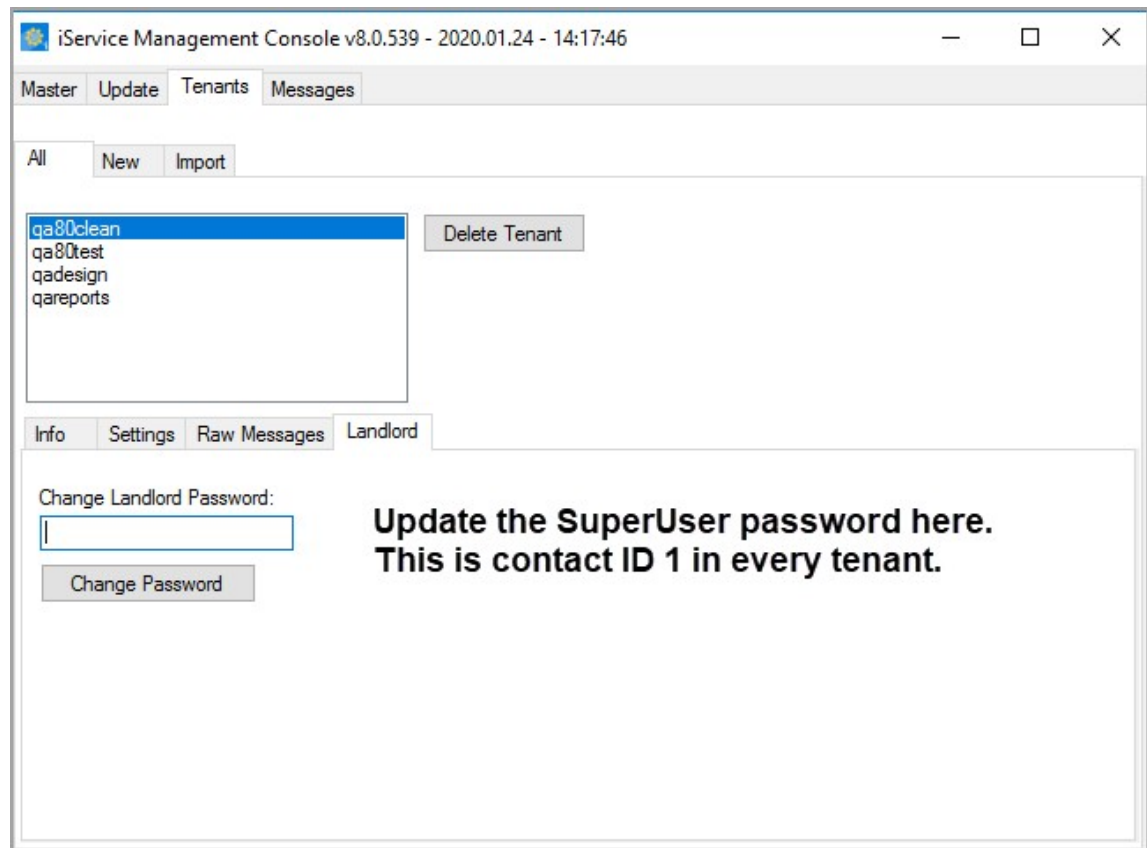
The Extract Failed Raw Message button is used to extract only the raw messages in a tenant database that failed processing. iService retrieves email and saves it into the tenant database, and then a separate process decodes them and creates interactions. From time to time, messages will be received without proper formatting and iService will be unable to process them. If you see that messages are failing processing in the Mail Monitor, you should extract them using this button for analysis.



The Raw Messages tab for extracting messages

Landlord Tab

The settings tab displays the number of agents licensed to use the tenant within the Agent Limit and Chat Agent Limit text boxes. The number of agents can only be modified if you have an unlimited user site license.

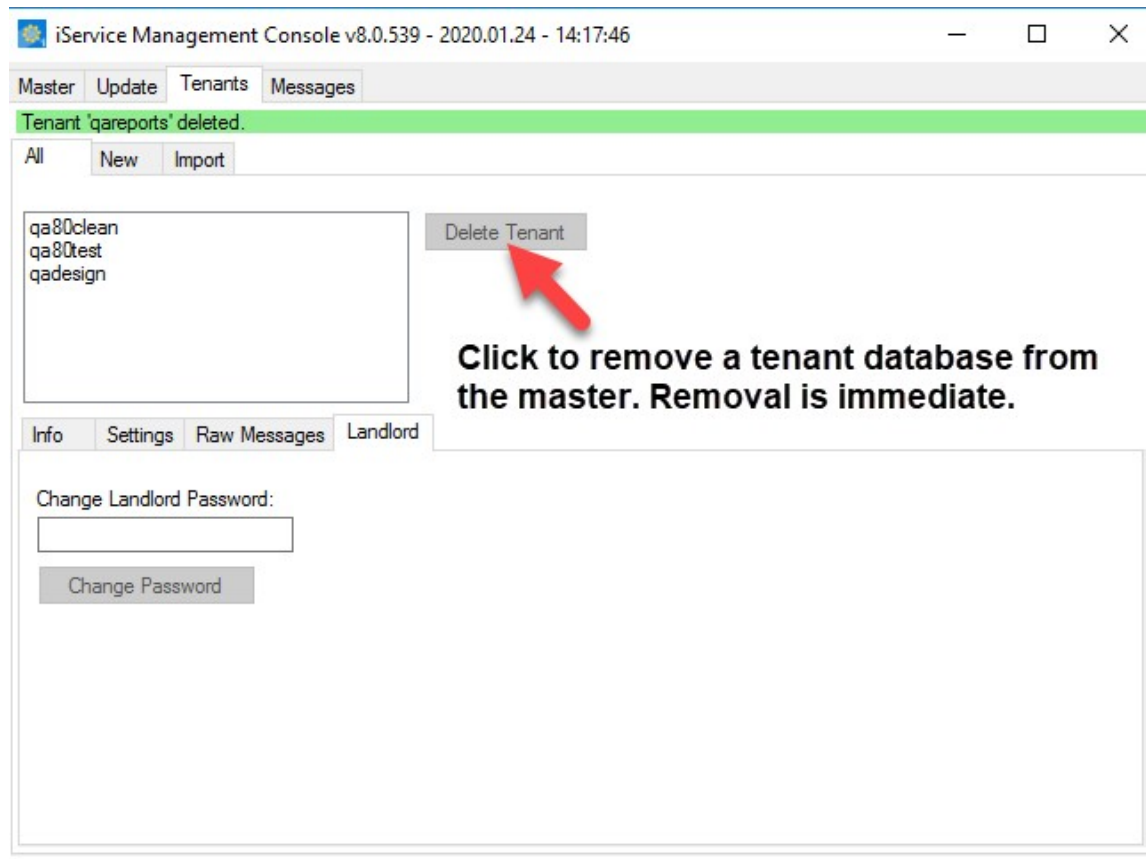


The Landlord tab in the Management Console

Deleting Tenants

When a tenant is deleted within the Management Console, its entry in the Master Database is removed. The Tenant database will remain in SQL Server until it is physically removed. To remove The tenant from the master database, click on the Tenants tab, click on the tenant you desire to remove, and then click “Delete Tenant.”

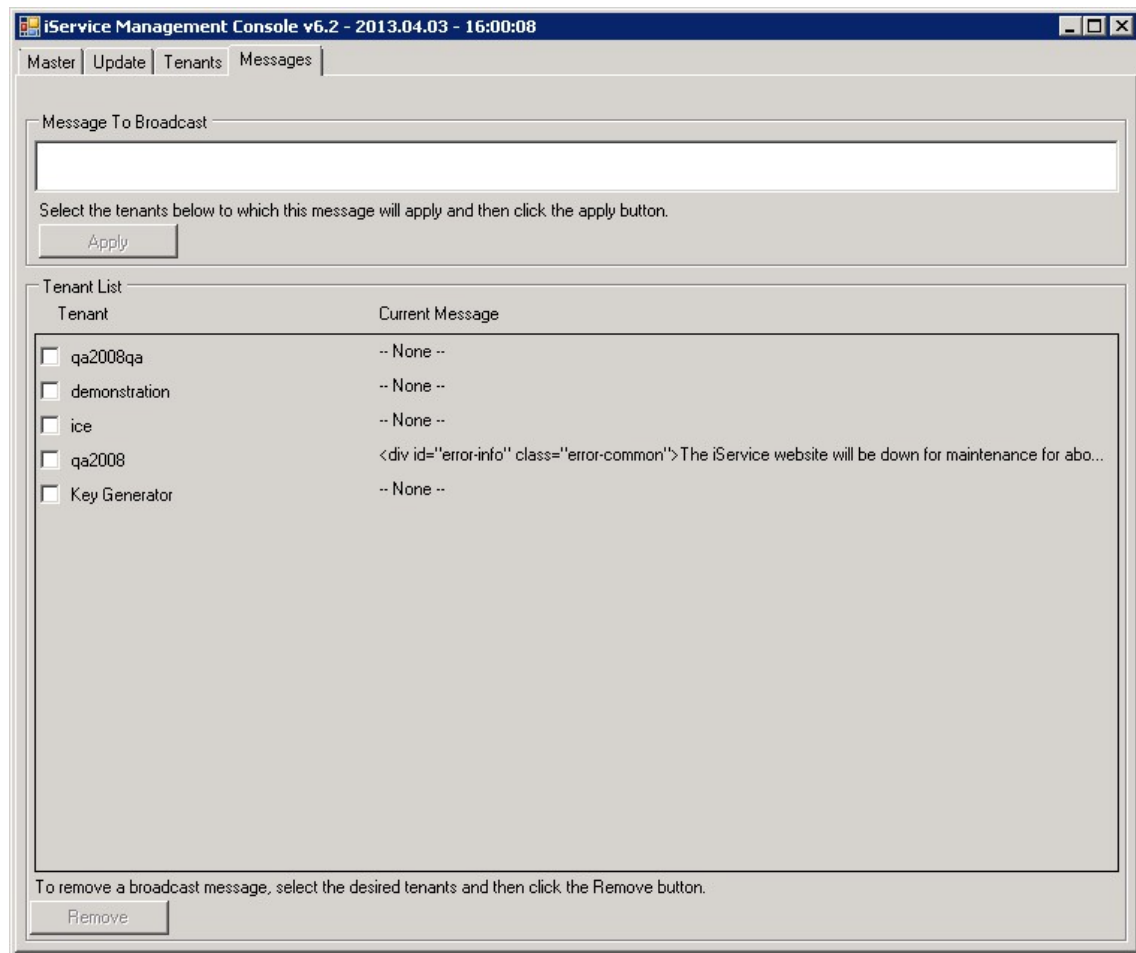
You must use SQL Server tools (e.g., SQL Server Management Studio) to permanently delete the database from SQL Server. The management console simply removes its references from the master database.



Deleting a tenant database

The Messages Tab

The messages tab is used to push messages to the logged in agents' browser window. These messages will appear at the top of the iService browser window. Messages can be plain text or any valid HTML that can be included in the BODY section of an HTML document.



Message to Broadcast

Enter your message that is to be broadcast to agents in the Message to Broadcast text area at the top of the screen. Select the tenants to receive the message from the Tenant List, and then click Apply to push the message to all agents within those tenants. The message will remain on the top of the agent's screen until it is removed. To remove the message, select the tenants from the Tenants list and click the Remove button.

You can enter plain text, such as:

The iService system will be down for maintenance at 10pm Central Time for

Or, you can enter an HTML version like the following:

<div>The iService system will be down for maintenance at 10pm</div>

Custom CSS For Styling Messages

iService (beginning with 6.1) includes a CSS style sheet that contains styles and images to make it easier to publish more informative messages. The four styles are Warning, Info, OK, and Caution. Examples of the HTML for these and their actual display are shown below.

Warning

```
<div id="error-warning" class="error-common">The iService system
will be down for maintenance at 10pm Central Time for approximately 10
minutes.</div>
```



The iService system will be down for maintenance at 10pm Central Time for approximately 10 minutes.

Info

```
<div id="error-info" class="error-common">The iService system will be
down for maintenance at 10pm Central Time for approximately 10
minutes.</div>
```



The iService system will be down for maintenance at 10pm Central Time for approximately 10 minutes.

OK

```
<div id="error-ok" class="error-common">The iService system will be
down for maintenance at 10pm Central Time for approximately 10
minutes.</div>
```



The iService system will be down for maintenance at 10pm Central Time for approximately 10 minutes.

Caution

```
<div id="error-caution" class="error-common">The iService system
will be down for maintenance at 10pm Central Time for approximately 10
minutes.</div>
```



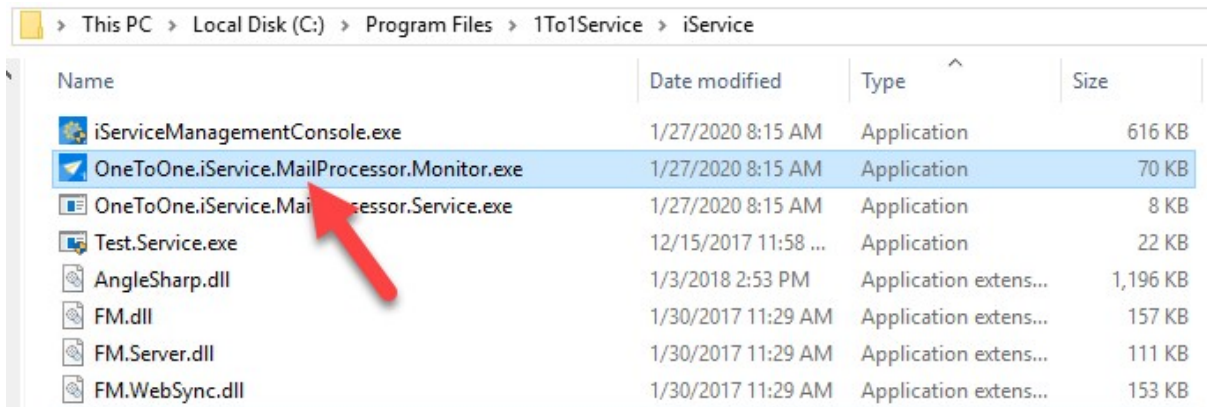
The iService system will be down for maintenance at 10pm Central Time for approximately 10 minutes.

3.2 Mailpopper Monitor

The iService MailPopper Monitor is used to monitor the status of the mail processor and other threads within the iService application. The mail

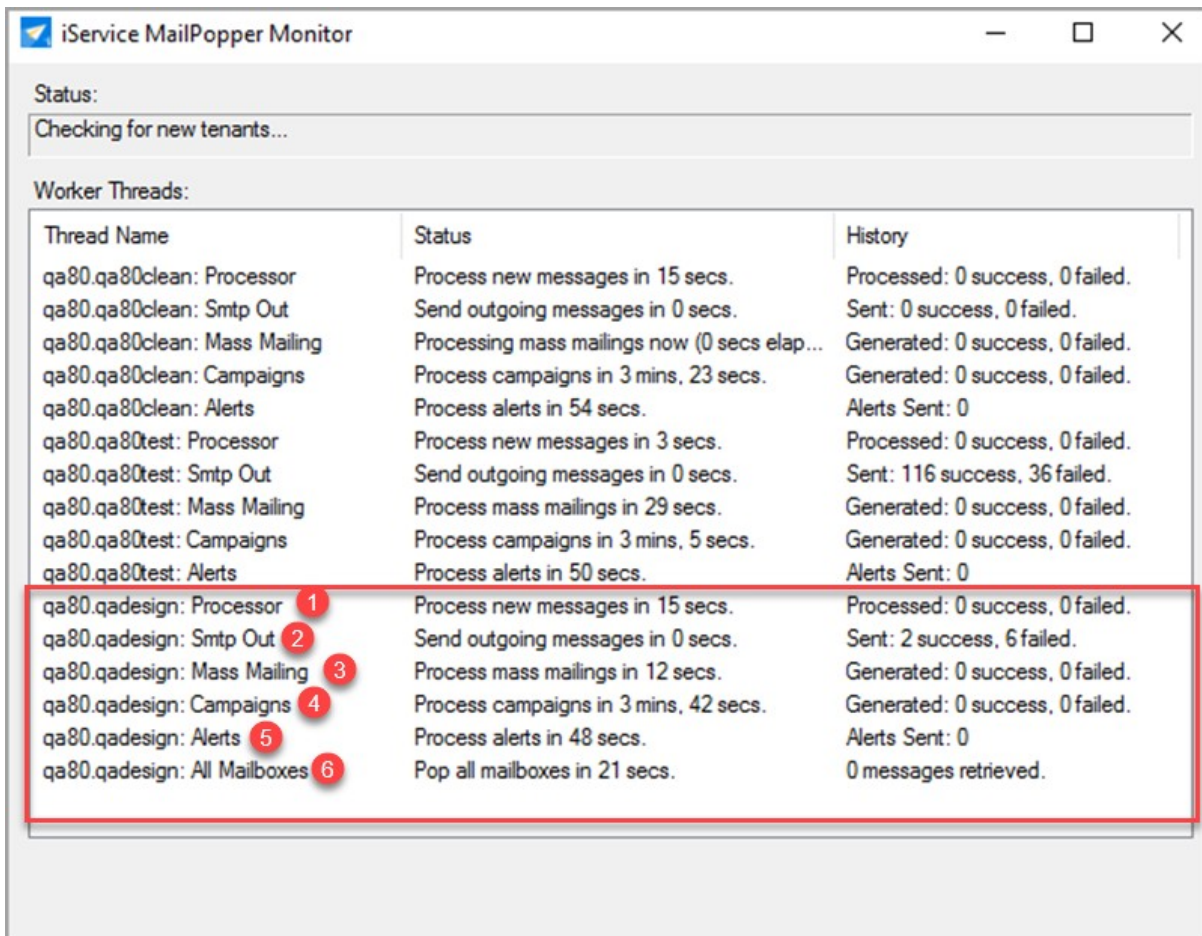
processor runs as a Windows Service that checks tenant mailboxes, retrieves those messages, and gets the messages into iService appropriately. The monitor provides a real-time view of the status of mailboxes including the number of messages being processed.

By default, its executable is located in the same directory as the iService Mail Popper service and iService Mail Monitor (the default location is C:\Program Files\1To1Service\iService).



The location of the MailPopper Monitor

When opened, the monitor provides details for every tenant in the installation as shown below.



Thread Name	Status	History
qa80.qa80clean: Processor	Process new messages in 15 secs.	Processed: 0 success, 0 failed.
qa80.qa80clean: Smt Out	Send outgoing messages in 0 secs.	Sent: 0 success, 0 failed.
qa80.qa80clean: Mass Mailing	Processing mass mailings now (0 secs elap...	Generated: 0 success, 0 failed.
qa80.qa80clean: Campaigns	Process campaigns in 3 mins, 23 secs.	Generated: 0 success, 0 failed.
qa80.qa80clean: Alerts	Process alerts in 54 secs.	Alerts Sent: 0
qa80.qa80test: Processor	Process new messages in 3 secs.	Processed: 0 success, 0 failed.
qa80.qa80test: Smt Out	Send outgoing messages in 0 secs.	Sent: 116 success, 36 failed.
qa80.qa80test: Mass Mailing	Process mass mailings in 29 secs.	Generated: 0 success, 0 failed.
qa80.qa80test: Campaigns	Process campaigns in 3 mins, 5 secs.	Generated: 0 success, 0 failed.
qa80.qa80test: Alerts	Process alerts in 50 secs.	Alerts Sent: 0
qa80.qadesign: Processor 1	Process new messages in 15 secs.	Processed: 0 success, 0 failed.
qa80.qadesign: Smt Out 2	Send outgoing messages in 0 secs.	Sent: 2 success, 6 failed.
qa80.qadesign: Mass Mailing 3	Process mass mailings in 12 secs.	Generated: 0 success, 0 failed.
qa80.qadesign: Campaigns 4	Process campaigns in 3 mins, 42 secs.	Generated: 0 success, 0 failed.
qa80.qadesign: Alerts 5	Process alerts in 48 secs.	Alerts Sent: 0
qa80.qadesign: All Mailboxes 6	Pop all mailboxes in 21 secs.	0 messages retrieved.

The MailPopper Monitor Screen

For each tenant, the monitor displays the following information. History is retained for the number of messages retrieved, sent, and processed until the mail popper service is restarted, at which time the count returns to zero.

1. **Processor** – After messages are retrieved and saved in the iService database (see All Mailboxes below), a mail processor thread is run to convert the raw message into an interaction. This line shows the number of emails that were successfully converted into interactions, and the number of emails that failed processing. Emails that failed processing can be extracted from the [Management Console](#)⁷⁷.
2. **SMTP Out** – Outbound email processing (SMTP) is managed separately from mail popping. The SMTP thread takes messages from the SMTPOUT table and delivers them to the appropriate SMTP mail server for

delivery. Messages that failed SMTP processing are shown as mailbox errors in the iService web interface (Settings - Tenant - Mailboxes page and the Buitin Forms - SMTP OUT Monitor page). See the User Guide for more details.

3. **Mass Mailing** – The mass mailing process selects the appropriate contact based upon list and criteria specified, and generates the SMTP message to be delivered. These messages are saved in the SMTPOUT table and then processed by the SMTP Out thread. See the User Guide for more details.
4. **Campaigns** – Campaigns function identically to Mass Mailings, but are based on a series of predefined message that process based upon a schedule. See the User Guide for more details.
5. **Alerts** – The Alerts thread monitors your iService tenant and takes actions, such as sending notification messages to managers for unresolved questions.
6. **All Mailboxes** - The status of mail popping is listed for all mailboxes within each tenant. the All Mailboxes line will not be shown if mail popping is disabled for the tenant.

Maintenance Tasks

4 Maintenance Tasks

From time to time, there are various maintenance tasks you need to perform on your iService installation to keep it optimized. Most of these relate to your database. This chapter highlights the most common.

4.1 Purging Interactions

Interactions may need to be deleted in an iService tenant. For example, in some cases sensitive data like credit card numbers must be removed for compliance reasons. Or, your database might grow large and require you to purge interaction data to reduce its size.

iService contains a stored procedure to delete interactions, but it cannot be accessed from the web interface and must be run by a system administrator or DBA with access to execute the `iServiceInteractionsDelete` stored procedure in the tenant database.


Notes:

- Deleting an Interaction is unrecoverable and removes all relational data connected to it, including:
 - Audit trails of the interactions status and properties
 - If interaction was sourced from an email, all related raw email and post-processed email data
- Other interactions in the same thread are NOT deleted. For example, if a customer emails in sensitive data and an agent replied to them with the original email quoted, both the customer email AND the agent reply must be deleted separately.
- If the parent of a child interaction is deleted, the child's grandparent interaction will be inherited as its parent, recursively. If the parent of a child interaction was the thread root, the child interaction and its siblings will all become thread roots.

The procedure simply takes one argument: the interaction ID

Example:

```
EXECUTE [dbo].[iServiceInteractionsDelete] 123456
```

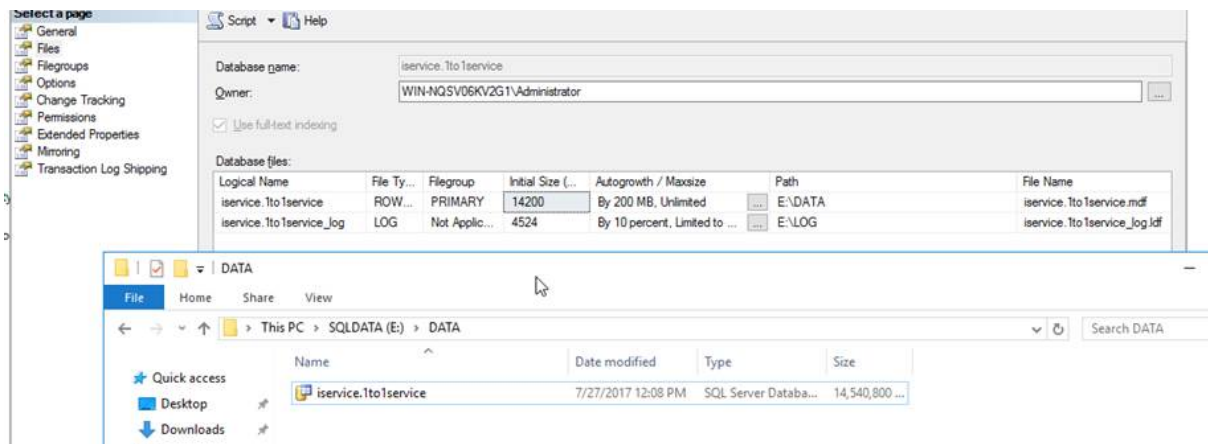
 iService includes a stored procedure to mass delete interactions, but you must first construct a query to derive the list of interactions to purge. For assistance with performing a mass delete on your database please contact One-to-One Service.com at support@1to1service.com.

4.2 Purging Contacts

The iService database has a stored procedure (`iServiceContactsDelete`) for deleting contacts from your tenant database. Contacts are associated with many records and many tables in the database, and `iServiceContactsDelete` manages the removal process. The stored procedure accepts the ID of a single contact.

4.3 Moving Raw Messages Table

In our example, the original iService filegroup is located on the E: drive, and we'll be moving FilesRawMessages to the lower-tier F: drive. Our tenant database is around 14.5 GB:



A query for the FilesRawMessages clustered index size tells us that about 4.5GB of space is allocated to it:

```
USE [iService.iService]
GO
SELECT i.[name] AS IndexName
,Sum(s.[used_page_count]) * 8 AS IndexSizeKB
FROM sys.dm_db_partition_stats AS s
INNER JOIN sys.indexes AS i ON s.[object_id] = i.[object_id]
```

```

        AND s.[index_id] = i.[index_id]
WHERE i.NAME = 'PK_FilesRawMessages'
GROUP BY i.[name]
ORDER BY i.[name]
GO

```

Results		Messages
	IndexName	IndexSizeKB
1	PK_FilesRawMessages	4438624

First, create a new filegroup in the tenant database. We'll call it LOBdata.

```

USE [master]
GO
ALTER DATABASE [iservice.1to1service] ADD FILEGROUP [LOBdata]
GO

```

Add a new file located on the F: drive, inside the new LOBdata filegroup. Earlier we noted that FilesRawMessages was 4.5GB, so we're going to make the initial size 5120000KB to hold it all.

```

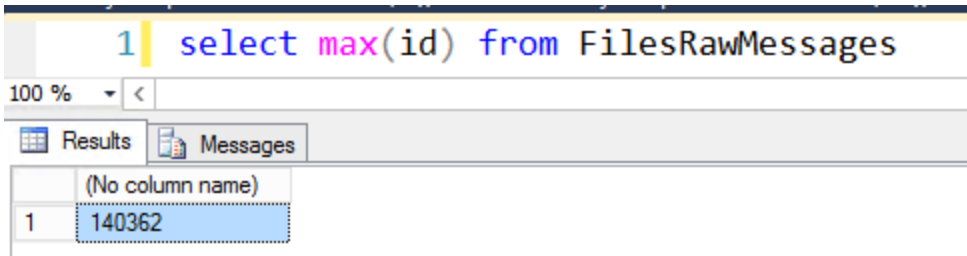
USE [master]
GO
ALTER DATABASE [iservice.1to1service] ADD FILE (
    NAME = N'iservice.1to1service_LOBdata'
    ,FILENAME = N'F:\DATA\iservice.1to1service_lobdata.ndf'
    ,SIZE = 5120000 KB
    ,FILEGROWTH = 204800 KB
) TO FILEGROUP [LOBdata]
GO

```

Next, we're going to partition the FilesRawMessages table, with the first partition residing in the LOBdata filegroup and actually containing all the rows in the table. The second partition will contain no rows and will reside in the original Primary filegroup. This will allow us to later re-index the table, moving the data to the LOBdata filegroup while keeping the database online.

We need to find what the highest row ID currently is in the FilesRawMessages table so that we can create a RIGHT partition range

much higher than it, so that any new rows created during the move also are within range:



Our highest ID is 140362, so a right range of 200000 should be enough. Create the partition function:

```
USE [iservice.1to1service]
GO
CREATE PARTITION FUNCTION rawmessagemove (INT) AS RANGE RIGHT
FOR
VALUES (200000);
```

Then apply the partition to the FilesRawMessages table, with partition #1 mapped to LOBdata, and partition #2 mapped to Primary:

```
USE [iservice.1to1service]
GO
CREATE PARTITION SCHEME rawmessagemove AS PARTITION rawmessagemove TO (
    lobdata
    ,[primary]
);
```

Next, we'll perform the actual move of the data, by dropping and recreating the indexes of FilesRawMessages onto the new filegroup. Let's get a list of the indexes we'll be working with:

```
USE [iservice.1to1service]
GO
SELECT dbschemas.[name] AS 'Schema'
    ,dbtables.[name] AS 'Table'
    ,dbindexes.[name] AS 'Index'
    ,indexstats.alloc_unit_type_desc
    ,indexstats.avg_fragmentation_in_percent
    ,indexstats.page_count
FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, NULL) AS indexstats
INNER JOIN sys.tables dbtables ON dbtables.[object_id] = indexstats.[object_id]
INNER JOIN sys.schemas dbschemas ON dbtables.[schema_id] = dbschemas.[schema_id]
INNER JOIN sys.indexes AS dbindexes ON dbindexes.[object_id] = indexstats.[object_id]
    AND indexstats.index_id = dbindexes.index_id
WHERE indexstats.database_id = DB_ID()
    AND dbtables.[name] = 'FilesRawMessages'
ORDER BY indexstats.avg_fragmentation_in_percent DESC
```

	Schema	Table	Index	alloc_unit_type_desc	avg_fragmentation_in_percent	page_count
1	dbo	FilesRawMessages	UQ__FilesRaw__43B15FA02EF4C8F4	IN_ROW_DATA	1.69753086419753	648
2	dbo	FilesRawMessages	PK_FilesRawMessages	IN_ROW_DATA	0.701402805611222	998
3	dbo	FilesRawMessages	PK_FilesRawMessages	LOB_DATA	0	554081

There are 3 indexes we'll need to move - the row data clustered index, the LOB data clustered index, and the unique unclustered index. As an aside, you might want to take note of the fragmentation shown, and compare it to the fragmentation after the move is complete.

First we'll drop and recreate the row data clustered index on the new partition (this may take a few minutes, and mail won't be processed during this time if doing it in OFFLINE mode):

```
USE [iservice.1to1service]
GO
CREATE UNIQUE CLUSTERED INDEX PK_FilesRawMessages ON dbo.FilesRawMessages ([ID] ASC)
WITH (
    DROP_EXISTING = ON
    ,ONLINE = OFF
) ON rawmessagemove(ID)
```

Next we'll drop and recreate the lob data clustered index, on the new filegroup. This will move all the LOB data (this again may take a few minutes, and mail won't be processed during this time if doing it in OFFLINE mode):


```
USE [iservice.1to1service]
GO
CREATE UNIQUE CLUSTERED INDEX PK_FilesRawMessages ON dbo.FilesRawMessages ([ID] ASC)
WITH (
    DROP_EXISTING = ON
    ,ONLINE = OFF
) ON [lobdata]
```

Finally, we'll do the same for the unclustered 'unique' index. Note that the name of the index will be different on your tenant. This should be quick compared to the other 2, and can be done ONLINE.

```
USE [iservice.1to1service]
GO
CREATE UNIQUE INDEX UQ__FilesRaw__43B15FA02EF4C8F4 ON dbo.FilesRawMessages ([hash] ASC)
WITH (
    DROP_EXISTING = ON
```

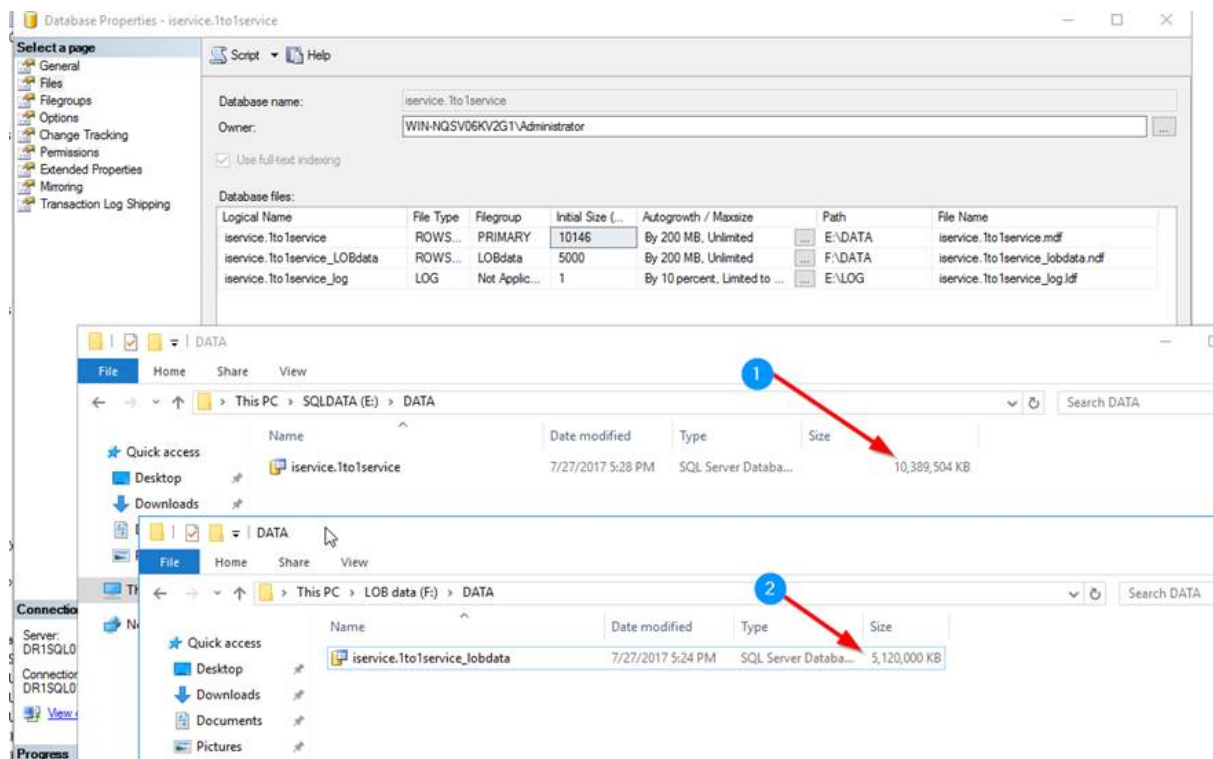
```
,ONLINE = ON
) ON [lobdata]
```

That's it! Your FilesRawMessages LOB have been moved to the other storage device. Now that your LOB data has been moved, depending on your future expected growth patterns you may or may not want to shrink the database to free up the newly unallocated space.

 **Note that this is completely optional, and there are some very negative consequences for doing so, one of which is fragmenting your freshly created indexes.**

In this example, we'll do a database shrink simply to demonstrate that the FilesRawMessages table was moved to the new storage.

1. Our original file in the primary filegroup on E: has been reduced from 14.5GB to around 10GB without FilesRawMessages.
2. The new file in the LOBdata filegroup on F: holds the 4.5GB of FilesRawMessages data.



The screenshot displays the SQL Server Enterprise Manager 'Database Properties' window for 'iservice.1to1service'. The 'FilesRawMessages' table is listed in the 'Database files' table, showing it has been moved to the 'LOBdata' filegroup on F:.

Logical Name	File Type	Filegroup	Initial Size (KB)	Autogrowth / Maxsize	Path	File Name
iservice.1to1service	ROWS...	PRIMARY	10146	By 200 MB, Unlimited	E:\DATA	iservice.1to1service.mdf
iservice.1to1service_LOBdata	ROWS...	LOBdata	5000	By 200 MB, Unlimited	F:\DATA	iservice.1to1service_lobdata.ndf
iservice.1to1service_log	LOG	Not Applic...	1	By 10 percent, Limited to ...	E:\LOG	iservice.1to1service_log.ldf

The Windows File Explorer windows show the following files:

Name	Date modified	Type	Size
iservice.1to1service	7/27/2017 5:28 PM	SQL Server Databa...	10,389,504 KB
iservice.1to1service_lobdata	7/27/2017 5:24 PM	SQL Server Databa...	5,120,000 KB

This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

Troubleshooting

5 Troubleshooting

This chapter describes some common issues we've seen for on-premise users.

5.1 Unable to Generate a Temporary Class

Server Error in '/' Application.

*Unable to generate a temporary class (result=1).
error CS2001: Source file 'C:\WINDOWS\TEMP\f7ni3e2f.0.cs' could not be found
error CS2008: No inputs specified*

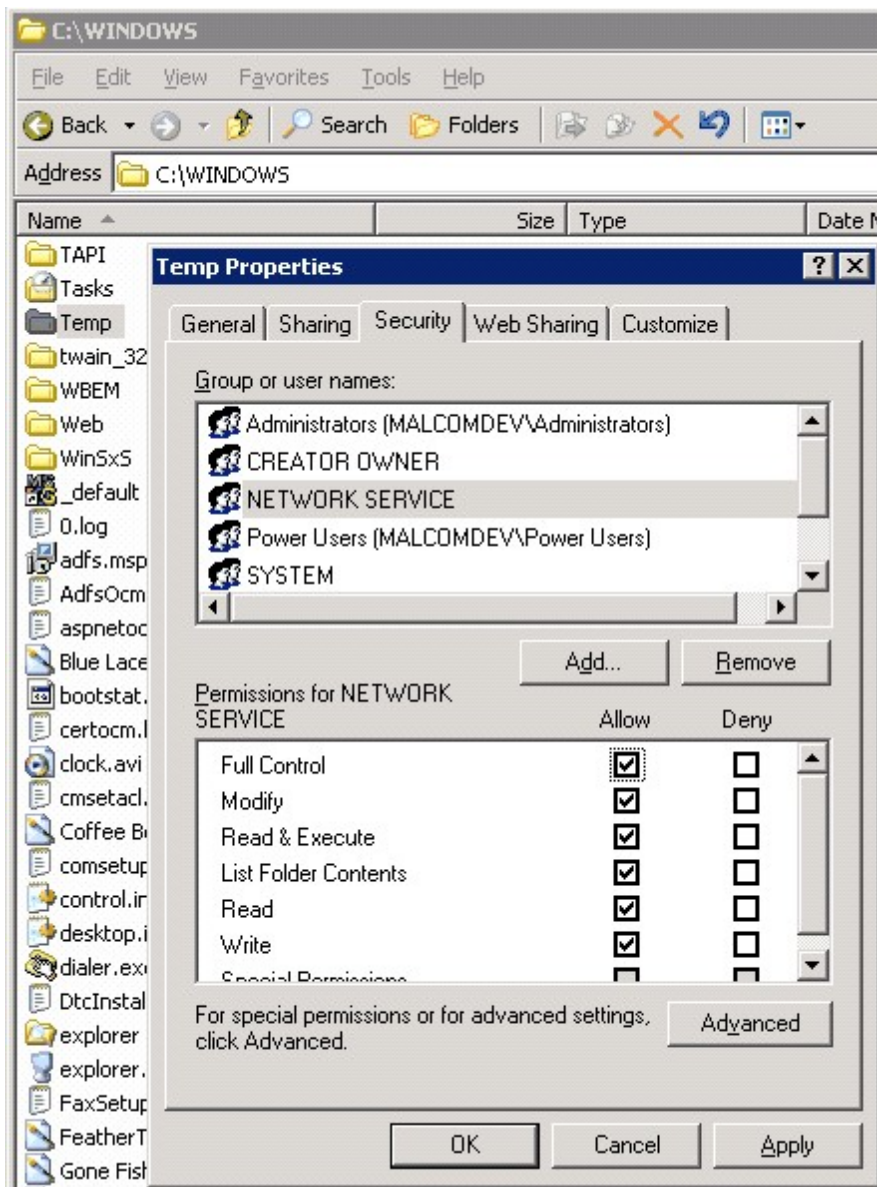
Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.InvalidOperationException: Unable to generate a temporary class (result=1).
error CS2001: Source file 'C:\WINDOWS\TEMP\f7ni3e2f.0.cs' could not be found
error CS2008: No inputs specified

Click to expand

In this case, the problem is that the web server cannot pre-compile the ASP.NET objects for your web site because it lacks the rights to the C:\WINDOWS\TEMP directory.

Add the NETWORK SERVICE identity to the Security properties for this folder as above, and give it full rights.



Once you have done this, click OK and try the web site again (to ensure that your browser does not show cached data, it is recommended that you close your browser completely and then reopen it before trying the web site again).

5.2 Messages Not Leaving iService

The iService.MailPopper Windows service must be running for the system to send and receive email. So, if your system is not sending or receiving email but there are no error log files, this would be the first thing to check.

When the mail processor is not running all mail will be queued until it is turned back on. As soon as you restart the service all queued mail will be delivered and all mail waiting to be popped in your monitored mailboxes will be processed.

Step 1 - Confirm your mail popper service is running (usually on the iService web server)

Step 2 - Look in the [Tools - SMTPOut Monitor](#) to see if messages are being rejected from the mail server

Step 3 - Look in [Segment Segments - Mailboxes](#) for errors on mailboxes

5.3 Checking table sizes in iService

From time to time, its important to evaluate the size of the iService database and the number of records in certain tables. For instance, if you are planning to run an interaction purge you will want to know how big your tables are.

The query below returns details about the tables within iService and can be helpful in understanding where your database is growing. It outputs the size of all tables in a database. You should run this query against the iService tenant database, not the iService Master database.

```
SELECT
    t.NAME AS TableName,
    p.rows AS RowCounts,
    SUM(a.total_pages) * 8 AS TotalSpaceKB,
    SUM(a.used_pages) * 8 AS UsedSpaceKB,
    (SUM(a.total_pages) - SUM(a.used_pages)) * 8 AS UnusedSpaceKB
FROM
    sys.tables t
INNER JOIN
    sys.indexes i ON t.OBJECT_ID = i.object_id
INNER JOIN
    sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id =
p.index_id
INNER JOIN
    sys.allocation_units a ON p.partition_id = a.container_id
WHERE
    t.NAME NOT LIKE 'dt%'
```

```

        AND t.is_ms_shipped = 0
        AND i.OBJECT_ID > 255
GROUP BY
    t.Name, p.Rows
ORDER BY
    t.Name

```

5.4 iService Error Log Files

iService generates log files when various errors occur. These log files can be generated from either the iService Web Services or the iService MailPopper service.

Web Service Log Files

Log files generated by the iService web services are written into the same directory as the iService Web Site. These files have two types of extensions: `successfulretry` and `log`. Log files with the `successfulretry` extension indicate that an initial attempt at a web service failed, but a second attempt was successful. These can usually be ignored.

Log files that have the `.log` file extension indicate a web service failure that was not successful on retry. These errors should be investigated as they indicate some type of problem with your installation.

MailPopper Service Log Files

Log files generated by the iService MailPopper Windows Service are recorded in the same folder as the service files. The default path for the web service is `\Program Files\1to1Service\iService`.

5.5 Incorrect URL for Web Services

If you see a 404 error within your log files, this often indicates that the path specified in your master database for the `iService.asmx` files is incorrect. You can check this by opening the master database for your installation and inspecting the `Tenants` table. Ensure the url listed for the iService Web Services is correct and accessible.

	ID	tenantName	webServiceUrl	timeLastSmt...	smtpSendGuid	timeLa
▶	1	qa2008qa	http://qa2008qa/iService.asmx	2014-05-09 06:...	NULL	2014-0

You must be able to load this URL in your browser and get a list of the iService web services.

- 4 -

404 error 97

- E -

Extract failed raw messages 77

- I -

Incorrect URL for Web Services 97

- L -

log files 97

- M -

MailPopper Service Log Files 97

- T -

The Messages Tab 66

- W -

Web Service Log Files 97