

Migrating from EM 9.7.x to EM 9.8

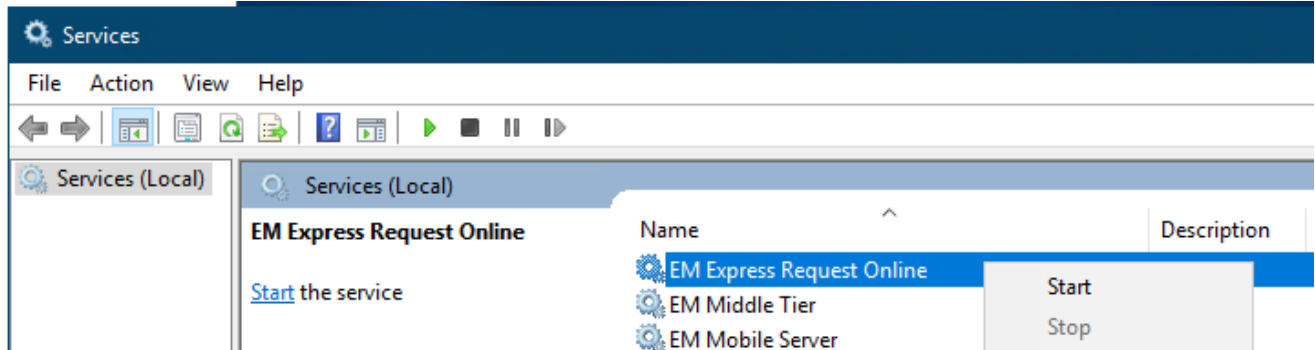
Overview

The release of 9.8 begins a new chapter for the Express Maintenance ecosystem. This will mark the release of our 2nd generation mobile application based on our new technology stack. This is a migration and you will need the **Install** installers: **EMInstall.exe**, **EMRESTServerInstall.exe**, and **ERInstall.exe**. Do not use the **update** versions of these installers. Make a backup of your current directory. Either copy it to a Temporary directory or create a zip file. All three installers will need to be ran as administrator. You can optionally make a manual backup of all the databases that will be migrated. **If you are on 9.6.8.2 or older**, please contact tech support for assistance. You will need to upgrade your database in stages. You can still proceed with installing the new REST Node.js Middle tier. Directions are below.

Express Request Online Running with EM 9.8.1

There is a new version of the Express Request Online server (ERO) installed only with the **9.8.1 EM Installer**. As the ERO application will soon be obsolete, it no longer gets installed automatically. Follow these steps to update an existing ERO installation.

1. Complete the EM 9.8.1 Installation using EMInstall.exe (not the update version).
2. Open the Services application and locate the EM Express Request Online service. If it is already running, use the right-click popup menu to Stop the service. Leave the Services application open. You will need it again in step 4.



3. Using File Explorer, copy the ERO application from the installation folder **C:\ExpressTech\WorkFiles\9.8\OtherFiles\EMExpressRequestOnline.exe** to the location where you installed EM. The default location is **C:\ExpressTech\EMExpressRequestOnline**.
4. In the Services application right click on the EM Express Request Online service and Start the service.
5. Run ERO to ensure it can communicate with the newly installed service.

Conversion Steps for EM

1. **If you are on 9.6.8.2 or older**, please contact tech support for assistance. You will need to upgrade your database in stages.
2. Download and run the EM 9.8 installer. **Do not auto-start EM. Migrating from an earlier version is considered a new install.**
3. Run any "Fixtltl" database script provided by MAS tech support.
4. Find and run the ExpressUpgradeAssistant from the install folder.
 - a. It will update your database to 9.8.0.
 - b. It will upgrade your encryption to a new latest and greatest algorithm.
 - c. It will make sure you have an Employee Record = 0 for "Unknown" and an Employee Record = 1 for "System Admin". These will be required for the new APIs and future features to work properly. If it needs to, it will relocate any existing record using those record numbers/ids.
 - d. Set up new tables for new and future enhancements.
5. If you use an INI file for your connection information, you will need to run ExpressConfig. There are new options and the system will automatically re-encrypt your data. If you have distributed this INI file to others, you will need to do so again. If you used command line options, you don't need to do anything.
6. If you need to install and register the Email DLL, you will need to download and run the EMClientInstall.exe. This installer will install just the supporting files and check to make sure you have a supported version of the ODBC driver installed.

Other Important Changes to 9.8

We have changed the way we store and process scripts in 9.8

- 9.8 does not have the DB scripts to update from older versions. You have to use the **Express Upgrade Assistant** to update your database to **9.8.0**.
- You will find a new directory in the "ExpressTech" folder: **WorkFiles**. This folder is where we will store files needed by the installer. These are accumulative, do not delete the folder or files.

- There is a new startup sequence. The login screen has been updated and there is a new “ScriptRunner” to process database updates. This is faster and easier for us to provide DB updates and it has error handling so updates are always processed properly or the process fails.
- The old system for updating the database has been in place for years and can fail without error. We have found that about 1/3 of the database we do support work for are not properly up to date. We offer a free service to analyze your current database schema and create an updated script to make sure it is up to date. Please contact tech support to discuss this process.

Node.js Application Server

Our new technology stack is based on Node.js which runs on both Windows Server and Linux. You will need Windows Server 2016 or newer and you can find the instructions below. You can also install the Node.js server on almost any version of Linux, but we will provide support for Ubuntu Server. You can find the instructions for Ubuntu Linux Server in the section after the Windows Server Install instructions. Once you have installed the “Application” server (Node.js and supporting products), we will have instructions on how to install the “Business” server on Windows and Linux in the last section of the document. The Business Server is the capabilities written by MAS to support and work with our EM ecosystem.

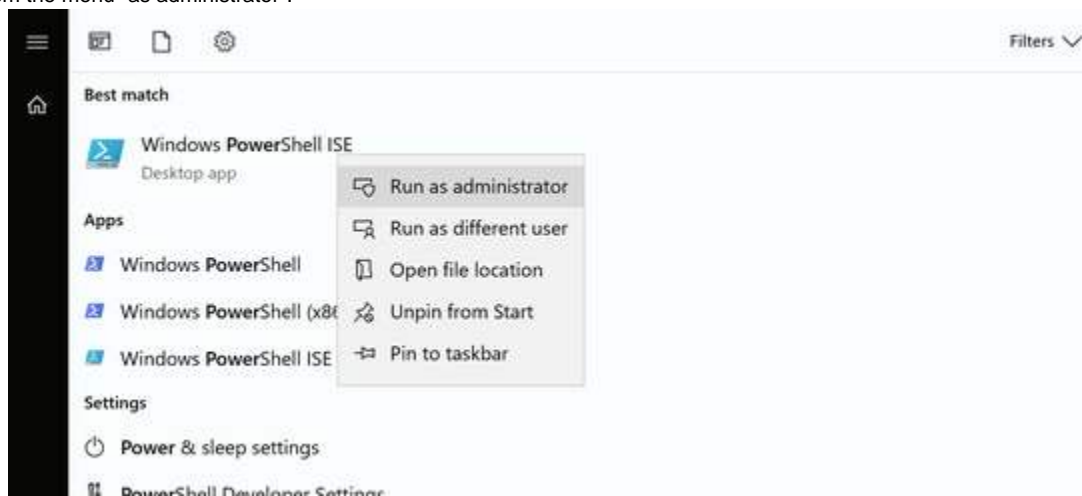
Node.js is the fastest-growing deployment platform for servers in the world. Node.js uses a non-blocking threading model and is one of the fastest servers in the world. It also runs JavaScript and TypeScript natively. Two of the most popular programming languages in the world (what companies are using for **new projects**).

Node.js, Express.js, and PM2 work fine on both Windows Server and Linux. Linux is a true server and these products are easier to install and run better on Linux. Window Server is a dual-purpose server and runs both in Server and User modes, with its default mode being User mode. When installing the Node.js Application server on Windows Server, extra steps need to be taken to have it run as a “server” application. If you plan on deploying the system so it is available on the internet, outside your corporate firewall, we highly recommend Nginx as a pass-through web server. Nginx is designed to run on Linux.

It likely sounds like we are trying to talk you out of using Windows Server. That is not the case, but we also want you to know that if you are looking for high-capacity capability (250+ concurrent connections at one time), we would highly recommend Linux. Many will only need 25 - 50 concurrent connections running inside the corporate firewall and Windows Server will handle this workload fine.

Installing Node.js Application Server On a Windows Server

Run the installer **EMRESTServerInstall.exe** “As Administrator”. This installer includes the EM Windows Services. If you have them installed, these will need to be shut down for the installer to run properly. We provide a script to streamline the installation process on Windows. Open PowerShell from the menu “as administrator”.



- Change the directory to **C:\ExpressTech\WorkFiles\node-install-util** directory.
- To run the setup script: **.\setup-npm-pm2-WS.ps1**

You may get a warning that you can not run scripts. You can turn off this “protection” by running the “Bypass” command line below in Powershell. Then run the script again. You can turn the protection back on by running the “Default” command below.

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Default
```

You can ignore the npm WARN on the global directive. Even if you do use the new directive, it still warns you. You also may want to reboot the server so all the environment changes are picked up properly.

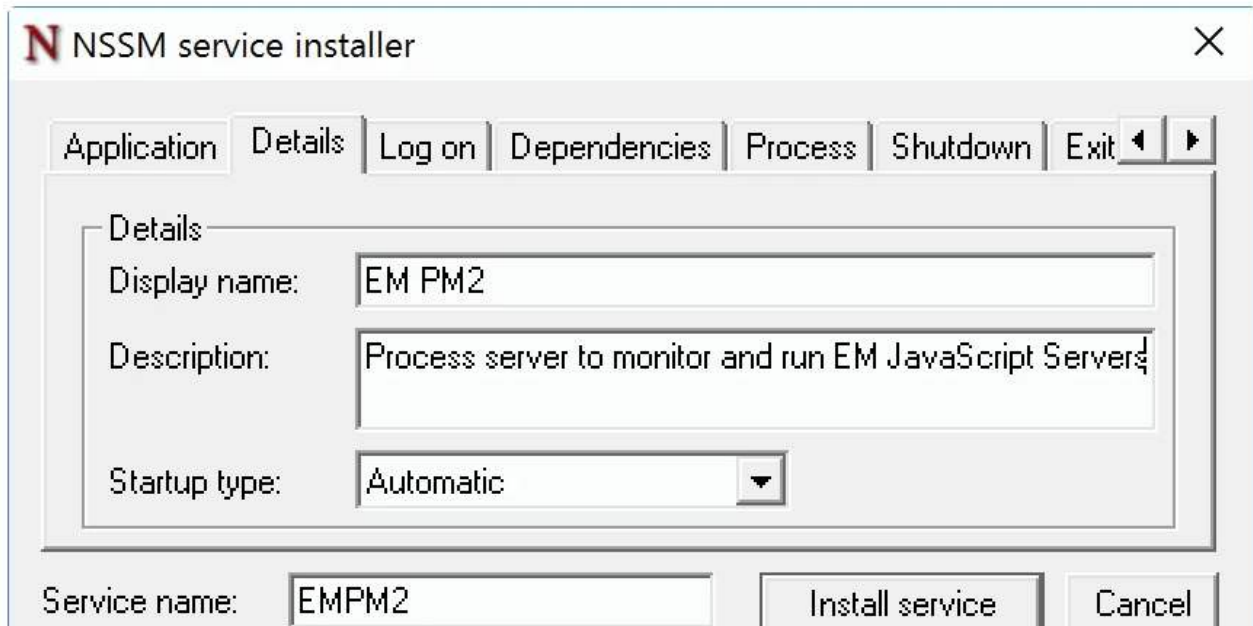
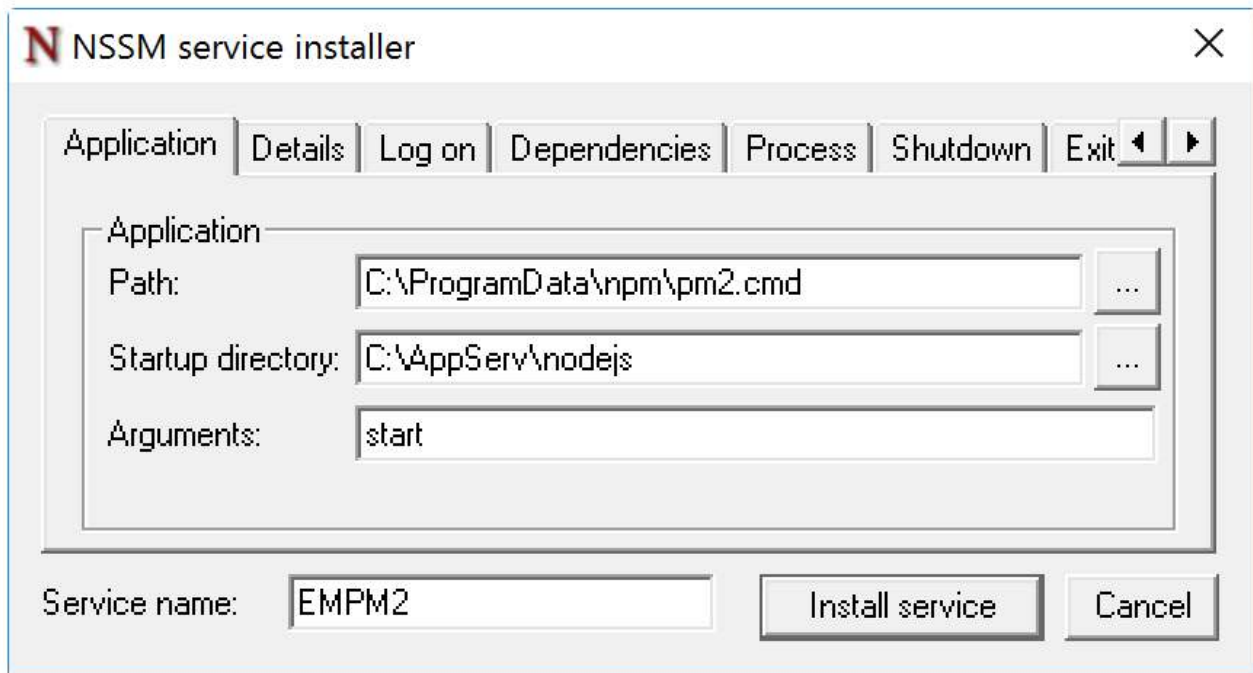
What we have accomplished so far:

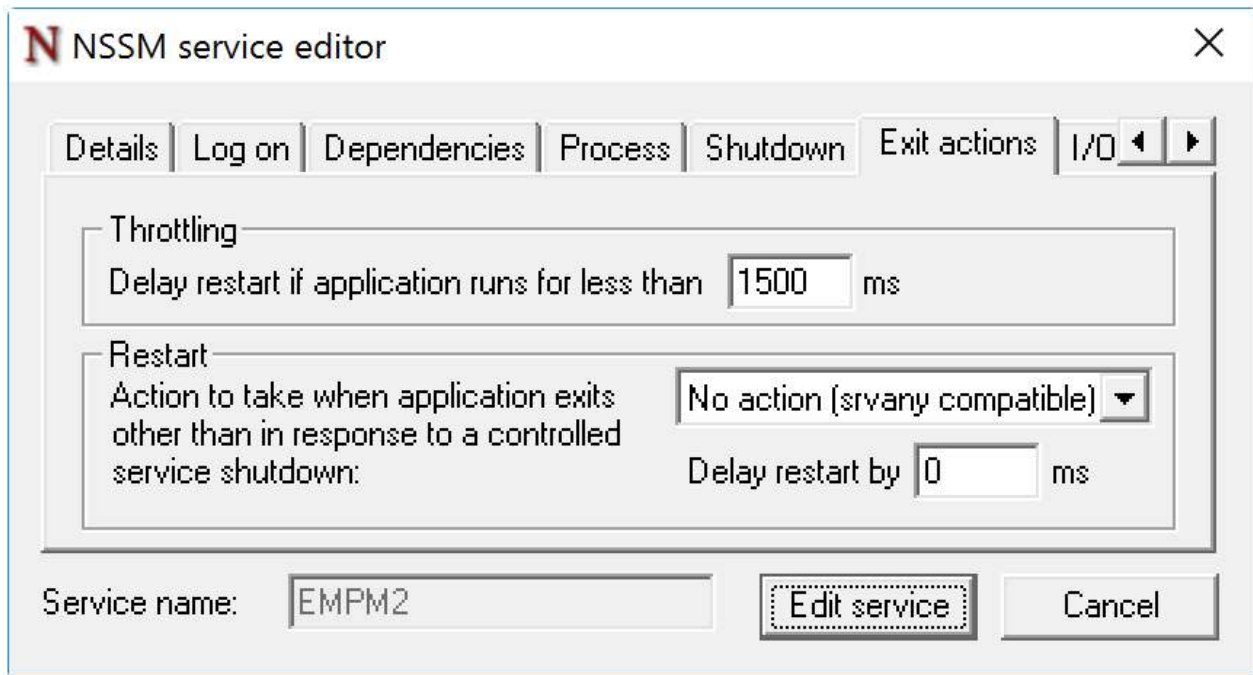
- Installed Node.js
- Installed PM2 and Log Management Module
- Configure folders and files to run as Administrator/System (required to run properly as a service)

Installing PM2 as a Service

We are using the open-source project NSSM to run PM2 as a Windows Service. You can find the commands to manage it here. Let's install pm2 as a service. First, let's stop the pm2 program by running "pm2 kill" in the command prompt. When setting the Display name it is important to use the name EM PM2 as shown below.

- Open the Node.js command prompt "As Administrator" (if not already open)
- Run: nssm install EMPM2
- A configuration screen will pop up. Fill in the Application and Details tabs below.





- Go to the **Exit** action tab and change the **Restart** setting to **No Action**.
- Install the service.
- You will need to start the service manually or restart the server (recommended).

Updating PM2 Startup Option for NSSM

If you need to update the service, you can use the following command.

```
NSSM edit EMPM2
```

These infrastructure items will not need to be installed again. If the service has started, turn off the service using the **Local Services** utility. Additional steps need to be completed and will be easier to complete without the service running.

Installing Node.js Application Server On Linux

We use Ubuntu Linux LTS as our production server OS. If you are looking for basic support from us on your installation and configuration, you will need to be running Ubuntu. Linux and Node.js are a very popular combination for a Web Server and an API REST Server. There are lots of resources on how to set up this software combination on Red Hat and Open SUSE Linux stacks.

Linux servers by their nature are non-visual first (vs Windows Servers being a user operating system first and a server software second). The traditional way to install application software like this is to stage the installation in one place and then copy the folders and files from the staging computer to the production server.

Installing Node.js and PM2

We will be installing Node.js 16. There are still some incompatibilities with some of the components and libraries that we are using. But we plan to update to Node.js 18 in the near future. Open a terminal window and run the following commands:

```
cd ~
sudo curl -sL https://deb.nodesource.com/setup_16.x -o nodesource_setup.
sh
sudo bash nodesource_setup.sh
sudo apt install nodejs
node -v
```

The result of these commands is downloading the install file, adding this to your configuration and your local package cache, installing node16.x, and verifying the version installed.

Now let's update npm to the latest version

```
sudo npm install -g npm@latest
npm -v
```

And install pm2

```
npm install -g pm2
pm2 -v
```

Register the pm2 log manager and configure it for 1GB log files.

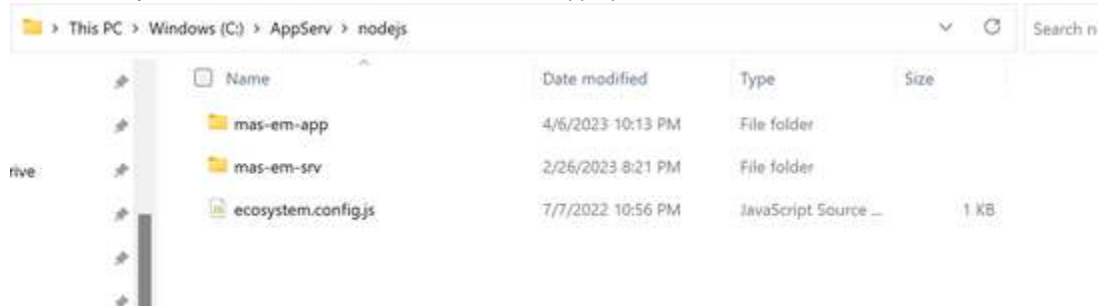
```
sudo pm2 start
sudo pm2 install @jessety/pm2-logrotate
sudo pm2 set @jessety/pm2-logrotate:max_size 1M
```

Installing the Business Server

Windows Server

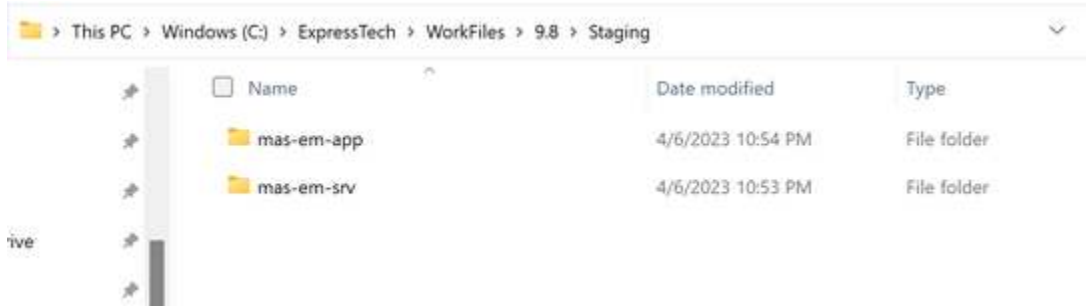
For Windows Server, application-related data is stored in the C:\ProgramData directory. All the applications and configuration information needed by the application server software is stored here in sub-directories. Business Applications are usually stored in the Programs directory or AppServ directories. We will be using the AppServ directory, specifically C:\AppServ\nodejs

The installer will automatically install the REST and Web server into their appropriate directories.



Note: The **mas-em-app** folder will be in a future release and might not be included in this release

When you use the “Updater”, you can choose between installing it in the production directories or in a staging folder.

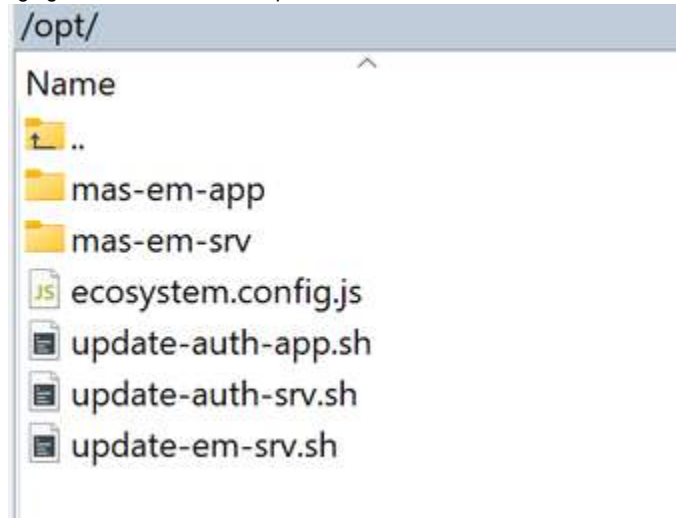


With PM2 configured to auto-start on a file change, the “staging” method of copying files into the directory is often used.

See the section “Setting up EM-SRV Components” and “Setting up EM-SRV Database Configuration” below to finish the installation.

Ubuntu Linux Server

For Debian-based Linux servers, the /opt directory. Linux users should always use the “EMRESTServerUpdate.exe” installer. This installer runs on Windows, stores the files in the staging folder, and then use a product like WinSCP to transfer the files to the Linux server.



Setting up EM-SRV Components

Whether you are deploying on Windows or Linux, you will need to install the components and libraries used by the business server. The application (em-app) has the components installed as part of the file package.

- In a command prompt (as Admin for Windows), change to the application directory, **C:\AppServ\nodejs\mas-em-srv**
- Run **npm install** and on Linux, **sudo npm install**
- This will install the components. You don't need to run this command on minor releases unless the release notes specifically state to do so. You should update the components on major quarterly releases.

Setting up EM-SRV Database Configuration

To configure the DB connectivity, you will need to update the multi-tenant-edit.json file. This can be found in **C:\AppServ\nodejs\mas-em-srv** or **/opt/mas-em-srv** . Use a text editor to modify the file. There are three examples of how to do the connection configuration.

1. SQL Server on the same server as the Business Node server.
2. SQL Server on a different server. Note that “\” is a reserved character, so you need 2 to escape the character in JavaScript (MyServer\\MySQLInstance).
3. Maria DB server, we use this internally for our Authentication Server and will eventually offer it as an option to SQL Server.

```
[
  {
    "key": "EABC4BBF-E865-4E6F-BEB0-9D59007F1A22",
    "db": "mssql",
    "server": "DESKTOP-SQLSERVER",
    "port": "1433",
    "database": "MyTestDB",
    "username": "admin",
    "password": "ae88fgAW2x"
  },
  {
    "key": "EABC4BBF-E865-4E6F-BEB0-9D59007F1A22",
    "db": "mssql",
    "server": "MyServer\\MySQLInstance",
    "port": "",
    "database": "MyTestDB",
    "username": "admin",
    "password": "5sPyYR3N5wK"
  },
  {
    "key": "EABC4BBF-E865-4E6F-BEB0-9D59347F3B68",
    "db": "mdb",
    "server": "localhost",
    "database": "mas-auth-test",
    "username": "admin",
    "password": "WXyz123abcEFG789"
  },
]
```

The “key” is the Database Key assigned when the database is created and is unique across all databases in the EM ecosystem. There are several ways to retrieve the database key.

- From EM, Info menu tab, choose the Connection Info. Locate the Database Id.
- Run SQL Statement: **SELECT KeyValue FROM dbo.EM_Config WHERE ConfigId = 10000;**

You should set up a configuration for each database you want the Business Server to work with. The configuration file can handle as many databases, mixed types, and locations (servers) as you want. It can handle hundreds of databases and servers.

Once the database servers are configured, you will need (want) to encrypt the user name and password used to connect to the database. There is a built-in utility to accomplish this task.

- The utility looks for unencrypted server configurations within the file and encrypts them.
- The utility loops through the servers and tests each server connection (all of them, new and old).
- If any database fails, it will be reported and halt the process.
- If it completes successfully, you will get a message: **Processing Multi-Tenant file updates complete!**

There are two ways to call the procedure. Through a web browser or using curl (a command line utility for making HTTP calls).

- <http://localhost:4100/util/proctenant>
- `curl -v http://127.0.0.1:4100/util/proctenant`

The new middle tier should be up and running. You can test it by “pinging” on e of the test endpoints:

<http://localhost:4100/util/test>