July 2020

Enterprise Route Management

*Environment Setup and Configuration Guide*

**Table of Contents**

**Section Title Page**

[1.0 About This Document 2](#_Toc45712530)

[1.1 Purpose 2](#_Toc45712531)

[1.2 Authors & Contributors 2](#_Toc45712532)

[1.3 Revision History 2](#_Toc45712533)

[2.0 System Overview 3](#_Toc45712534)

[2.1 Prerequisites 3](#_Toc45712535)

[3.0 ArcGIS Enterprise 4](#_Toc45712536)

[3.1 Portal & Server 4](#_Toc45712537)

[3.2 Routing Services 4](#_Toc45712538)

[4.0 Middleware/Node 7](#_Toc45712539)

[4.1 Node.js 7](#_Toc45712540)

[4.2 IIS 7](#_Toc45712541)

[4.3 Forward Proxy 7](#_Toc45712542)

[4.4 Middleware URL 8](#_Toc45712543)

[5.0 Web Server 8](#_Toc45712544)

[5.1 IIS 8](#_Toc45712545)

[5.2 RoutePlanner URL 8](#_Toc45712546)

[6.0 Certificates 9](#_Toc45712547)

[6.1 Domain-signed 9](#_Toc45712548)

1. About This Document
   1. Purpose

This Environment Setup Guide serves to document steps for software installation and system configuration that is required before deploying the Enterprise Route Management system.

This document is based on an ArcGIS Enterprise 10.8 deployment.

Steps to deploy the custom ERM application components are covered in a separate document *ERM Application Deployment Guide.*

* 1. Authors & Contributors

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Role/Title | Email | Phone |
| Mike Nelson | Esri Release Manager | mnelson@esri.com | (909) 793-2853 x5202 |
| Michael Gayheart | Esri Scrum Master | mgayheart@esri.com | (909) 793-2853 x4426 |
| Ashley DePottey | Esri Tech Lead | adepottey@esri.com | (909) 793-2853 x1824 |

* 1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Editor |
| 5/22/2020 | 0.1 | Initial Draft v0.1 | Mike Nelson |
| 3/5/2020 | 1.0 | 10.7.1 initial version | Mike Nelson |
| 6/12/2020 | 2.0 | Updates for 10.8, various edits | Mike Nelson |
| 7/2/2020 | 3.0 | apply feedback from Dev deployment | Mike Nelson |

1. System Overview

There ERM environment is made up of 3 components:

* An ArcGIS Enterprise deployment with Portal and Server. (Referred to as “Enterprise” throughout this doc).
* A dedicated windows server that hosts the Enterprise Route Management API service. (“Middleware”)
* A web server that hosts the ERM web application. (“Web Server”)

It is recommended that these 3 components are placed on three separate machines, either cloud based or physical. These instructions assume 3 separate machines.

For test or sample environments these could be combined to use fewer machines.

These instructions assume standing up environment from scratch. Existing web server or ArcGIS Enterprise deployment can be used if meet requirements.

See *Appendix A Environment Checklist* for a list of basic tasks to complete.

* 1. Prerequisites
* Middleware server will use Windows Server 2019.
* The Middleware API will be exposed as a site through IIS. This site will need to be public facing for the application to access.
  + “Public” in this sense can be an intranet
* ArcGIS Enterprise version 10.7.1 or 10.8
  + There are slight paths differences. They are called out in subsequent instructions.
* Node JS version 12.18.0
  + Any 12.x version is acceptable.

1. ArcGIS Enterprise
   1. Portal & Server

ERM uses a base configuration ArcGIS Enterprise with Portal and Server. There is nothing custom required for ERM, so the core installation instructions can be used.

* [Installing Portal](https://enterprise.arcgis.com/en/portal/latest/install/windows/installing-portal-for-arcgis.htm)
* [Installing Server](https://enterprise.arcgis.com/en/server/latest/install/windows/welcome-to-the-arcgis-for-server-install-guide.htm)

If installing on same machine can also use ArcGIS Enterprise Builder. If putting on Azure or AWS instance, ArcGIS Cloud Builder can be used.

* 1. Routing Services

The ERM takes advantage of the Vehicle Routing Problem (VRP) within Network Analyst geoprocessing service. This can be deployed with StreetMap Premium (SMP) for ArcGIS.

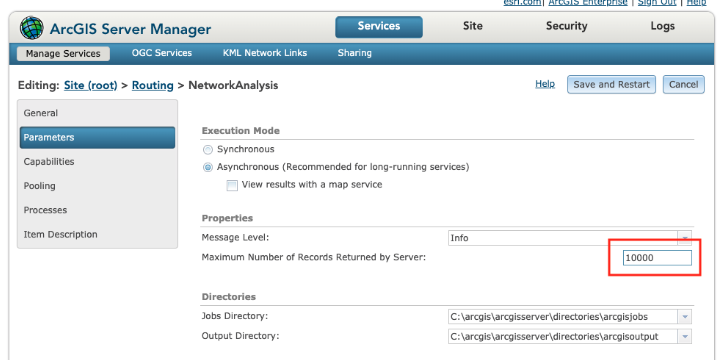
More information on publishing routing services can be found [here](https://enterprise.arcgis.com/en/server/latest/administer/windows/publishing-routing-services.htm) and [here](https://doc.arcgis.com/en/streetmap-premium/get-started/download-guide.htm).

1. On the Enterprise server, create a folder to hold the SMP data. make a folder.
   1. i.e. C:\\_data\SMP\HERE\North\_America\_2019
2. Copy and extract each zip from StreetMap Premium into your folder
   1. NA\_2019R4\_ClassicLocators.7z.001
   2. NA\_2019R4\_ClassicLocators.7z.002
   3. NA\_2019R4\_Documentation.7z
   4. NA\_2019R4\_FGDB.7z.001
   5. NA\_2019R4\_FGDB.7z.002
   6. NA\_2019R4\_NewLocators.7z
   7. NOTE: for the .002 files, you only need to extract .001 files using 7zip. It will automatically grab .002. Trying to extract .002 files directly will throw error.
3. Create a folder to hold services C:\\_data\RoutingServices\ServiceDefinitions
4. Use the publish routing services script included with Server to publish SMP.
   1. For 10.8, the script is located here: C:\Program Files\ArcGIS\Server\tools\PublishRoutingServices\routingservices\publishroutingservices.py
   2. For 10.7, the script is located here: C:\Program Files\ArcGIS\Server\tools\PublishRoutingServices\publishroutingservices.py
   3. Run the code sample below from an admin command prompt. Update paths and information where applicable.

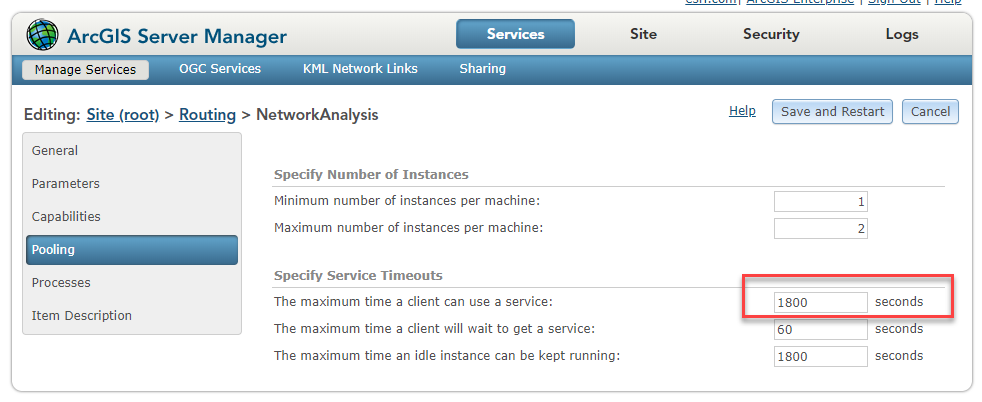
“C:\Program Files\ArcGIS\Server\framework\runtime\ArcGIS\bin\Python\envs\arcgispro-py3\python.exe” "C:\Program Files\ArcGIS\Server\tools\PublishRoutingServices\routingservices\publishroutingservices.py" -s <server name> -P <server name> -u <admin user> -p <password> -o C:\\_data\RoutingServices\ServiceDefinitions -n C:\\_data\SMP\HERE\North\_America\_2019\FGDB\StreetMap\_Data\NorthAmerica.gdb\Routing\Routing\_ND

* 1. The python.exe path might be different depending on where server was installed.
  2. For server name, use the fully qualified domain name of the Enterprise machine.
  3. For admin user and password, use an ArcGIS Server admin account.
  4. The NorthAmerica.gdb\Routing\Routing\_ND path will be different, depending on where you extracted your data.

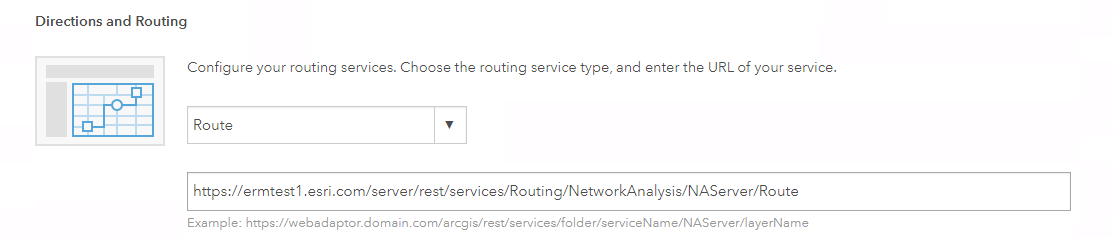
1. Open Server Manager and verify 4 published services in Routing folder.
   1. NetworkAnalysis (GP Service)
   2. NetworkAnalysis (Map Service)
   3. NetworkAnalysisSync (GP Service)
   4. NetworkAnalysisUtilities (GP Service)
2. Under Routing folder, open the NetworkAnalysis geoprocessing service.
3. On the Parameters page, verify the Maximum Number of Records returned by Server to at least 10000.



1. On the Pooling page, verify the Max time a client can use a service to at least 1800.



1. Click Save and Restart button.
2. Set the routing service in Portal
   1. Log into Portal as admin
   2. Go to Organization > Settings > Utility Services
   3. Under Directions and Routing set Route option and enter the URL for your Network Analysis service

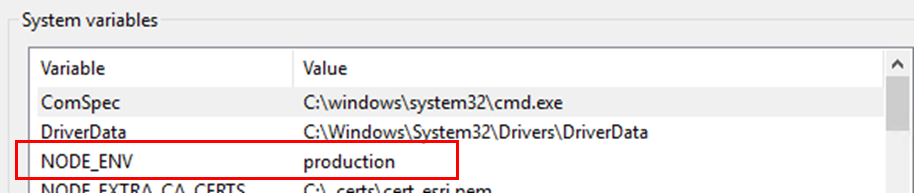


1. Middleware/Node

These steps will be performed on the machine hosting the ERM middleware API.

* 1. Node.js

1. Install version 12.18.0 64-bit of LTS from <https://nodejs.org/en/download/>
   1. Use all defaults in installer. Do not need to include extra tools option.
2. Add an environment variable NODE\_ENV. Set value equal to “production” or “development”, depending on which environment you are deploying.



* 1. IIS

When the middleware API is deployed, it will create a site IIS. These are precursor steps.

1. Enable IIS
2. Open IIS and browse to default web site
3. Choose Bindings
4. Select HTTPS and click Edit
5. Set your SSL certificate
6. Install both the [URL Rewrite](https://www.iis.net/downloads/microsoft/url-rewrite) and [ARR](https://www.iis.net/downloads/microsoft/application-request-routing) modules
   1. Forward Proxy

Need to setup IIS forward proxy so the ERM API can be accessed over the standard https port.

1. Open IIS
2. Create a new application under Default Web Site
   1. Alias = ermapi
   2. Use Default AppPool
   3. Set physical path to where ERM middleware API will be placed. Such as C:\arcgis\ERM
3. Set up SSL for this application using a proper CA or domain-signed cert
4. Create a reverse proxy rule that forwards requests for the ermapi context to port 8000:
   1. In IIS select ermapi app
   2. Go to "URL Rewrite"
   3. In action pane, select "Add Rule(s)"
   4. Select "Reverse Proxy"
   5. Set the Rewrite URL to [http://localhost:8000](http://localhost:8000/)
   6. Middleware URL

May need to work with your IT staff to configure a URL so the application can reach the API.

https://<your URL>/ermapi

1. Web Server

It is assumed the Web Server will be a separate machine from Enterprise and Middleware, possibly leveraging existing Web Server.

* 1. IIS

1. Enable ISS on machine.
2. Install the IIS module [URL Rewrite](https://www.iis.net/downloads/microsoft/url-rewrite).
   1. RoutePlanner URL

May need to work with your IT staff to configure a URL that users will use to hit the web planner app.

https://<your URL>/routeplanner

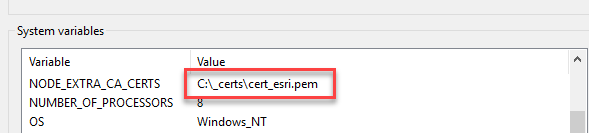
1. Certificates

Any required certificates for the servers to talk across the network will need to be available.

* 1. Domain-signed

By default, Node.js does not accept domain-signed certificates. To run this application against an ArcGIS Enterprise deployment with domain-signed certs, follow these steps:

1. Download the domain root certificate and transform it to a .pem file if needed. Save the certificate file somewhere that the node app and geoprocessing services can access.
   1. Place in a local folder such as C:\\_certs
2. Add an environment variable NODE\_EXTRA\_CA\_CERTS set to the path of this certificate



1. Repeat for both Middleware and Enterprise machines.
2. Environment Checklist

Here is a high-level list of tasks to complete to prepare the environment for deployment of the ERM application.

|  |  |  |
| --- | --- | --- |
| **#** | **Task** | **Complete** |
| 1 | ArcGIS Enterprise (Portal & Server) installed |  |
| 2 | Routing services from StreetMap Premium published |  |
| 3 | Portal configured to use routing services |  |
| 4 | Windows Server 2019 machine available for Middleware |  |
| 5 | Node.js installed on Middleware server |  |
| 6 | IIS deployed on Middleware server |  |
| 7 | Forward Proxy deployed on Middleware server |  |
| 8 | Middleware machine exposed with URL that app can hit |  |
| 9 | Web Server configured |  |
| 10 | URL available for routeplanner app |  |
| 11 | All certificates configured and available |  |