Last Update: May 24, 2023

Enterprise Route Management

*Application Deployment Guide*

**Table of Contents**

**Section Title Page**

[1.0 Overview 3](#_Toc133851775)

[1.1 System Prerequisites 3](#_Toc133851776)

[1.2 ERM Install Package 3](#_Toc133851777)

[2.0 Enterprise 5](#_Toc133851778)

[2.1 Portal Items 5](#_Toc133851779)

[2.1.1 Web Map for App 5](#_Toc133851780)

[2.1.2 Application Item 6](#_Toc133851781)

[2.1.3 Create Location Groups 7](#_Toc133851782)

[2.1.4 Users 7](#_Toc133851783)

[3.0 Business System Integration 8](#_Toc133851784)

[3.1 Sample BSI Service 8](#_Toc133851785)

[3.1.1 Setup BSI Files 8](#_Toc133851786)

[3.1.2 Publish BSI Service 8](#_Toc133851787)

[3.2 Sample Data 9](#_Toc133851788)

[3.2.1 Publish Sample Data 9](#_Toc133851789)

[4.0 ERM Feature Services 11](#_Toc133851790)

[4.1 Load Customer Data 11](#_Toc133851791)

[4.1.1 ERM\_Plan\_Defaults 11](#_Toc133851792)

[4.1.2 ERM\_Plan\_Template 12](#_Toc133851793)

[4.1.3 ERM\_Registry 12](#_Toc133851794)

[4.1.4 MDM\_Locations 12](#_Toc133851795)

[4.1.5 GPS\_Template 12](#_Toc133851796)

[4.2 Travel Modes 13](#_Toc133851797)

[4.2.1 Configure Travel Modes 13](#_Toc133851798)

[4.3 Publish Feature Services 14](#_Toc133851799)

[4.3.1 Batch Tool 14](#_Toc133851800)

[4.3.2 Manual Publish 15](#_Toc133851801)

[5.0 Create Dashboard Service 16](#_Toc133851802)

[5.1.1 Publish with Tool 16](#_Toc133851803)

[6.0 Apply Configuration 17](#_Toc133851804)

[7.0 ERM Middleware Server 18](#_Toc133851805)

[7.1 IIS & Forward Proxy 18](#_Toc133851806)

[7.2 Install ERM API Service 19](#_Toc133851807)

[7.2.1 Validate Install 19](#_Toc133851808)

[7.3 ERM API Log 19](#_Toc133851809)

[8.0 Route Planner 21](#_Toc133851810)

[8.1 Deployment 21](#_Toc133851811)

[8.2 Browser Support 21](#_Toc133851812)

[9.0 Updating ERM 22](#_Toc133851813)

[9.1 Updating Feature Services 22](#_Toc133851814)

[9.1.1 Data Updates 22](#_Toc133851815)

[9.1.2 Data Model Updates 23](#_Toc133851816)

[9.2 Updating Middleware 23](#_Toc133851817)

[9.2.1 Configuration 23](#_Toc133851818)

[9.2.2 Upgrade 23](#_Toc133851819)

[9.3 Updating Route Planner 24](#_Toc133851820)

[Appendix A Readiness Checklist 25](#_Toc133851821)

[Appendix B App Deployment Checklist 26](#_Toc133851822)

[Appendix C Web Map Template 27](#_Toc133851823)

[Appendix D Create Dashboard Template 29](#_Toc133851824)

[Add Additional Data Layers 30](#_Toc133851825)

[Create Template Dashboard 30](#_Toc133851826)

[Update Configuration 30](#_Toc133851827)

[API Config 31](#_Toc133851828)

[Web Config 31](#_Toc133851829)

[Appendix E Base Functionality Test 32](#_Toc133851830)

[Appendix F Document History 33](#_Toc133851831)

1. Overview

This Application Deployment Guide serves to document steps for deploying the components for the Enterprise Route Management system. The first sections cover an initial or clean install. See [Section 7](#Section7) for instructions and notes on modifying or upgrading an existing installation.

The Enterprise Route Management (ERM) application is made up of several components:

* A browser based ERM application (Route Planner) that users interact with
* An ArcGIS Enterprise deployment that hosts ERM data and services, and controls access to application (Enterprise)
* A dedicated server that hosts the ERM API (Middleware) service
* A geoprocessing service that allows ERM to integrate with the business system of record (Business System Integration or BSI)
  1. System Prerequisites

The following components for the EMR environment should already be set up. Depending on the environment these may be 3 separate machines, or some machines might be shared in development/test environments.

* Base deployment of ArcGIS Enterprise (Portal, Server and relational Data Store).
  + Including Street Map Premium install for Network services
  + If using Server 10.9.1, also need [VRP Patch](https://support.esri.com/en/download/7992)
* Server to host ERM middleware API.
* Web Server to host ERM Route Planner Application.
* Ancillary items installed or available, such as Node.js installation, certificate files.

The *ERM Environment Setup Guide* available in the [ERM GitHub document repository](https://github.com/EsriPS/enterprise-route-management) has instructions for preparing the environment.

See [Appendix A](#AppendixA) in this doc for a readiness checklist.

* 1. ERM Install Package

A release package will have been provided from Esri for deployment of the ERM components. It will contain the following directories:

* Dashboard service – files to publish GP service to create dashboards from plans.
* Deployment – files to aid in deployment of ERM.
* Integration service – file to publish a sample BSI GP service.
  + Use for testing and system validation. Only needed if not using customer BSI.
* Middleware – code files for ERM API service.
* Routeplanner – web application files.
* Scripts – code for publishing ERM API service.
* Services\_Sample – contains ArcGIS Pro project and sample data to publish to use with sample Integration services.
  + Only needed if using sample BSI.
* Services – contains ArcGIS Pro project, empty file geodatabases to load customer data into, and tools to publish ERM Feature Services.

It is recommended to extract the zip onto a machine that has ArcGIS Pro. From here, files will be used or updated and then copied to the different servers.

1. Enterprise

This section covers ERM items to be published to or created in the ArcGIS Enterprise deployment.

* 1. Portal Items

ERM requires a few items in Portal:

* Web Map
* Web Application
* Group for each Location
* General ERM Group if not sharing all items with Organization (Optional)

These items can either be shared with your entire organization, or you can create a main ERM group only for ERM items and users. If shared to organization, all users will have access to these items.

Recommended to create these items using same administration account that will be used to publish services later. Can also add additional tags such as “ERM” for organizational purposes. Only required tags are included in instructions.

It is also recommended to place all ERM items in a folder in Portal called ERM Items, or something similar, for easy reference.

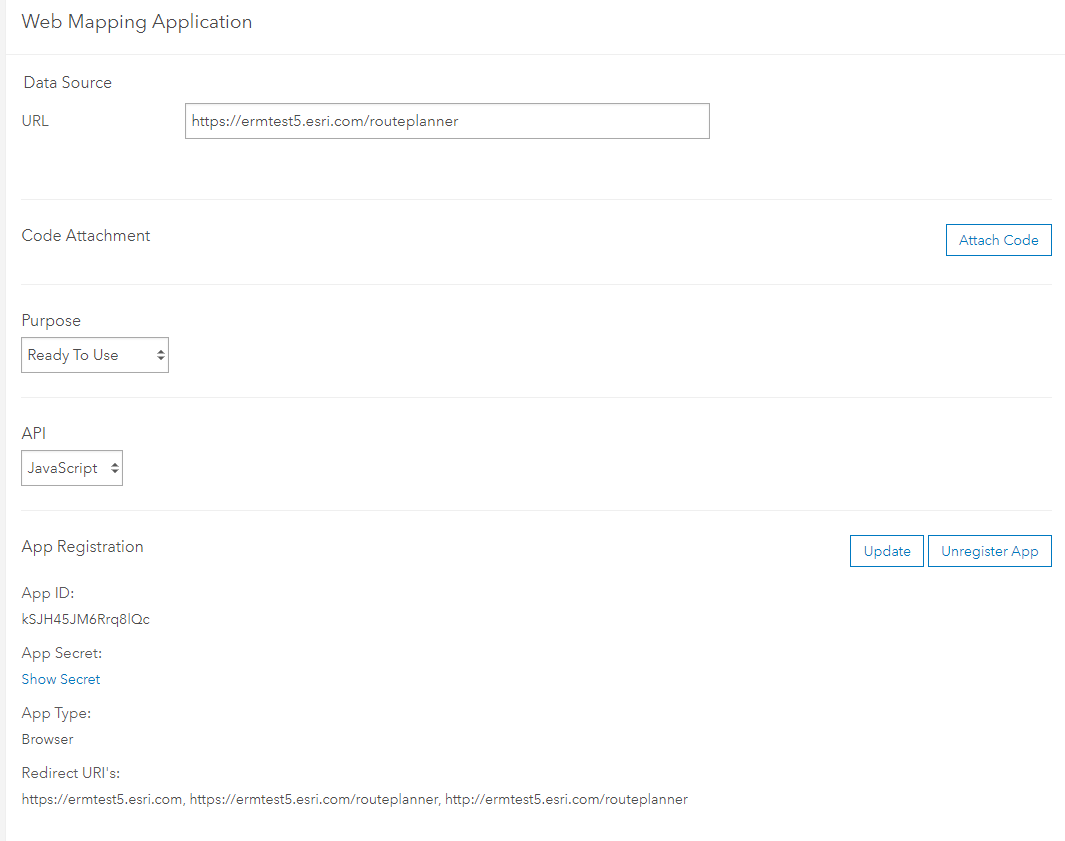
* + 1. Web Map for App

The ERM application needs a blank web map to point at. Create a new web map for the application to use. There is a later optional step for creating a template web map that would contain non-ERM layers. See [Appendix C](#AppendixC) for instructions.

1. Open an empty web map and choose Save As
   1. Detailed directions can be found in Esri online help [here](https://enterprise.arcgis.com/en/portal/latest/use/get-started-with-maps.htm).
2. Give a name such as “ERM default map”. Name not used by app.
3. Do not add any layers. Can leave Basemap as default, application configuration will set the value.
   1. Note that the very first time Route Planner is loaded, before any plans are created for a location, user will see this basemap.
4. Initial extent does not matter, application will define.
5. Share the web map with your Organization or a defined ERM group. After map is created save the item ID (found in the URL) to be used when configuring the Route Planner app (can store all IDs in same txt file for reference).
6. Open Settings of map and mark “Prevent item from being accidentally deleted.”
   1. Can optionally mark as Authoritative as well.
      1. Application Item

Create a web application item in Portal to register Route Planner.

1. From Content, choose New Item
2. Chose Application
3. Choose Web mapping and set the URL to what the Route Planner URL will be
   1. https://<your URL or machine name>/routeplanner
4. Title = RoutePlanner
5. Optionally place into a folder
6. Set any Tags for organizing.
7. Press Save to create item.
8. Open item details (after saving it should put you in Details by default)
9. Share the application item with your Organization or a defined ERM group
10. On Settings tab go to Web Mapping Application section
11. Set Purpose = Ready to Use
12. Set API = JavaScript
13. Click Register App
14. Add redirect URLs for your app with https and http.
    1. https://<your URL or machine name>/routeplanner
    2. http://<your URL or machine name>/routeplanner
15. Save the item ID found in the URL and the registered ID for the item.
16. Mark “Prevent item from being accidentally deleted.”
    1. Can optionally mark as Authoritative as well.



* + 1. Create Location Groups

A Create Groups tool is available in the services Pro project included with the release package.

1. Open the Pro project and log into your Portal.
2. From the ERM\_Utilities python toolbox, open the Create Groups tool.
3. Add the name of each location.
4. Run tool.
5. Groups will be created with following settings:
   1. A tag of the format "dispatch-location-xxx"(where xxx denotes the dispatch location. For example, "dispatch-location-OCC", "dispatch-location-COV", etc.)
   2. People in the organization can view group.
   3. Group Members can contribute content.
   4. Group members can update only their items.

You will also need to log into Portal and add users to the correct group(s) in the next section.

* + 1. Users

Users that will be accessing Route Planner application need to have accounts that meet these criteria:

* User account has either a Publisher or Administrator Role set.
* Users must be a member of Group for any Location that they plan.
* If using a main ERM group to share items rather than to the full Organization, User must be member of that group.

1. Business System Integration

A Business System Integration (BSI) service needs to be published that links ERM and the system of record for orders and collections. Each customer will be developing and deploying their own custom BSI service. The procedure and details for setting up the production BSI service are handled in separate documentation.

This document handles setting up a sample BSI service to use for testing and system validation. If not using the sample BSI and have your custom BSI service already setup, can skip to next section.

* 1. Sample BSI Service

A sample service can be set up to simulate a BSI service.

|  |  |
| --- | --- |
|  | * for these instructions the deploy location is set as C:\arcgis\ERM. This can be changed to another drive or folder on the server machine if needed. |

* + 1. Setup BSI Files

1. On the Enterprise machine, create the following folders:
   1. C:\arcgis\ERM\BSI
   2. C:\arcgis\ERM\Commit
2. From the extracted ERM release package zip, copy the jsonIntegration folder (found under integration service\data) into C:\arcgis\ERM\BSI on the Enterprise
3. In the python\config folder:
   1. Update the file path in both the GetPlanObjects.ini and JSONGetPlanObjects.ini to where you copied sample json (C:\arcgis\ERM\BSI\ jsonIntegration)
   2. Update the file path in CommitPlan.ini to match the created Commit folder (C:\arcgis\ERM\Commit)
   3. These files will get copied over after tool is published.
4. On the Enterprise machine, create an Environment Variable
   1. Name = NODE\_EXTRA\_CA\_CERTS
   2. Value = point to the .pem domain certificate
      1. Publish BSI Service
5. Open ArcGIS Pro and log into your Portal.
6. Open the Upload Service Definition tool from the geoprocessing pane.
7. Point to the BSI.sd file in the integration service\python folder.
8. Under the Override Sharing Properties section, check box to share with your Organization or ERM group.
9. Set any folder options for where to store service.
10. Run tool.
11. Open Server Manager.
12. Find where the folder where the tool is published. Will vary depending on where the Server installation is, but should be like C:\arcgis\arcgisserver\directories\arcgissystem\arcgisinput
13. Open your folder and drill down to the p20 folder. Depending on install and name used, should be like C:\arcgis\arcgisserver\directories\arcgissystem\arcgisinput\BSI\BSI\_Test.GPServer\extracted\p20
14. From the extracted python folder, copy the config folder into the p20 folder.
    1. This is the folder where you updated the .ini files.
15. Can open Server Manager and verify item is published and is started.
16. Can open Portal and verify item is shared correctly.
    1. Sample Data

If using the sample Business System Integration Service, you will need to publish the sample data that goes with it. This included in the release package will be a services\_Sample directory. The file geodatabases are populated with data to be used with the sample BSI. The publishing steps will be the same, but it is recommended to add a “\_Sample” or other suffix to the services to denote them from customer data that could be published later.

If only using sample data for initial install, can skip the Load Customer Data section.

* + 1. Publish Sample Data

1. From the services\_Sample folder, open the ERM\_services Pro project.
2. Connect to Portal as admin user being used to publish all items.
3. From Catalog View, choose Toolboxes and open the ERM\_Utilities toolbox.
4. Open the Publish ERM Data tool and fill in the parameters.
   1. Project File – is prepopulated. It assumes you are using the ERM\_services project you are in and that data lives in fgdbs folder.
   2. Maps with Data to Publish – choose all if this is initial install.
   3. Tags – add any tags you want to have on services. Optional.
   4. Service Suffix – if publishing sample data or a different version, can add a suffix to the end of service name. Optional
      1. If suffix is added, will need to manually update configuration files where applicable. Recommended to only use suffix if need to have multiple versions of the layers.
   5. Portal Folder to Publish to – Optional folder on Portal to store all layers in. Will create if does not exist.
5. Run tool.
   1. This creates a sd\_files\_temp folder in your project directory. If you need to run the tool again you will need to delete the folder first (tool will warn you).
6. Verify in Portal that the following items are created. There will also be a service definition file for each layer that can be deleted.
   1. ERM\_Plan\_Defaults feature layer
   2. ERM\_Plan\_Template feature layer
   3. ERM\_Registry table
   4. MDM\_Locations feature layer
   5. GPS\_Template feature layer
7. ERM Feature Services

There are several feature services that need to be published for ERM. Before publishing, the default feature classes will need to be populated with customer data. An ArcGIS Pro project is included with the release that contains maps for each feature service to be published.

The same user that was used to create Portal items in a previous section should be used to publish the feature services.

The Sample data included in the services\_Sample directory is to be used with the sample BSI. Directions are in the section for installing sample BSI service.

* 1. Load Customer Data

Included in the release package is a services folder that contains the file geodatabases and ArcGIS Pro project to publish.

This step can be done on any computer with ArcGIS Pro installed and access to the ERM Portal and Server.

1. From the extracted delivery directory, copy the services folder to machine with ArcGIS Pro. Or an accessible location for Pro.
2. From the services folder, open the *ERM\_services.aprx* Pro project included with release
3. Verify the following maps are available and have no broken links. If links are broken, point to the layers in the fgdbs folder.
   1. ERM\_Plan\_Defaults
   2. ERM\_Plan\_Template
   3. ERM\_Registry
   4. MDM\_Locations
   5. GPS\_Template

|  |  |
| --- | --- |
|  | * For all maps it is recommended to leave layers in same sequence that they are presented. The application expects the index value of some layers to be certain value. * These can be updated in the web app configuration file but leaving in default order will save from having to update those values. |

* + 1. ERM\_Plan\_Defaults

Update layers in the ERM\_Plan\_Defaults map with data for your locations. Core Pro tools can be used to import locations from other databases. See online help documentation [here](https://pro.arcgis.com/en/pro-app/latest/help/data/geodatabases/overview/import-data.htm).

* PointBarrierLibrary – add point features that act as a barrier (optional).
* DepotTemplate - add a point feature for each location.
* LineBarrierLibrary – add linear features that act as a barrier (optional).
* PolygonBarrierLibrary – add polygon features that act as a barrier (optional).
* ZoneTemplate – add polygon features that define zones (optional).
  + Zones are assigned to Routes here by filling in the RouteName attribute. Routes are unable to be assigned within Route Planner.
* BreakTemplate table – add break information for each route (optional).
* DispatchLocation table – add information for each location.
  + Information on populating the Travel Mode field is in [Travel Modes section](#TravelModes).
  + The index value for this layer is set in the ERM API configuration. If you change layer order, will need to update value.
* RouteTemplate table – add all available routes for all locations.
* SpecialtyNameTemplate table – enter information on available specialties (optional).
  + 1. ERM\_Plan\_Template

It is recommended to not adjust the ERM\_Plan\_Template map and to use a Template Web Map to alter symbology. See [Appendix C](#AppendixC) for instructions.

* + 1. ERM\_Registry

The ERM\_Registry map needs no updates. This is used by the application to store a list of all plans.

* + 1. MDM\_Locations

The MDM\_Locations are used to populate a pick list to select locations to move orders to. For example, if a central depot serviced several different yards. If not using this functionality, layer can be left blank.

1. Add point feature(s) for each location as needed.
2. Fill in attribution. Note that the displocname value should match the Depot name to relate the location to the Depot.
   * 1. GPS\_Template

No update needed. This layer would be where GeoEvent would be configured to put vehicle locations. This is optional functionality that can be configured, especially if using Workforce in conjunction with Route Planner.

* 1. Travel Modes

When Solving Routes, ERM will use Travel Modes that are configured on your Routing Service. These are set per location.

|  |  |
| --- | --- |
|  | * **Support for ERM\_Solve\_Parameters tables has been depreciated for ERM release 4.0** |

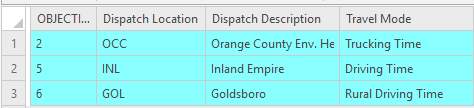
* + 1. Configure Travel Modes

In Portal, under Organization > Settings > Utility Services in the Directions and Routing section the available Travel Modes are listed.



ERM supports any Time-based mode. You can use the default modes or create your own. Existing modes can be duplicated and then adjusted to fit your needs.

In the ERM\_Plan\_Defaults map, the Travel Mode is set in the DispatchLocation table. The name needs to match the Travel Mode name exactly, including spaces.



* 1. Publish Feature Services
     1. Batch Tool

A tool is included with the ERM\_services Pro project for publishing.

1. Open the ERM\_services Pro project included with release that was used to populate the data.
2. Connect to Portal as admin user being used to publish all items.
3. From Catalog View, choose Toolboxes and open the ERM\_Utilities toolbox.
4. Open the Publish ERM Data tool and fill in the parameters.
   1. Project File – is prepopulated. It assumes you are using the ERM\_services project you are in and that data lives in fgdbs folder.
   2. Maps with Data to Publish –choose all if this is initial install.
   3. Tags – add any tags you want to have on services. Optional.
   4. Service Suffix – if publishing sample data or a different version, can add a suffix to the end of service name. Optional
      1. If suffix is added, will need to manually update configuration files where applicable. Recommended to only use suffix if need to have multiple versions of the layers.
   5. Portal Folder to Publish to – Optional folder on Portal to store all layers in. Will create if does not exist.
5. Run tool.
   1. This creates a sd\_files\_temp folder in your project directory. If you need to run the tool again you will need to delete the folder first (tool will warn you).
6. Verify in Portal that the following items are created. There will also be a service definition file for each layer that can be deleted.
   1. ERM\_Plan\_Defaults feature layer
   2. ERM\_Plan\_Template feature layer
   3. ERM\_Registry table
   4. MDM\_Locations feature layer
   5. GPS\_Template feature layer
      1. Manual Publish

If there are tool issues, or you just want to manually run through same steps, this procedure to follow:

1. Open the ERM\_services Pro project included with release that was used to populate the data.
2. Connect to Portal as admin user being used to publish all items.
3. Share each map as a separate hosted feature/table service.
   1. Open a map.
   2. Activate Share tab and select Web Layer > Publish Web Layer
   3. Enter a Summary and optionally add tags.
   4. Select "Copy all data" option in the Data section.
   5. Set "Feature" option in the Layer Type section.
   6. Share with Organization
   7. Select Configuration tab and click pencil icon for Feature Layer.
   8. Check Enabled Editing and Export Data options.
   9. Analyze the service.
      1. Can ignore template warnings.
   10. Publish service.
4. Create Dashboard Service

Route Planner can call a geoprocessing service to create a Dashboard from a template for that Plan.

This is optional functionality, if not used can leave the Dashboard items blank in the configuration file. You will also need to set the showDashboardControls value to false in the Route Planner configuration file to hide the Create Dashboard button.

* + 1. Publish with Tool

1. Open ArcGIS Pro and log into your Portal.
2. Open the Upload Service Definition tool from the geoprocessing pane.
3. Point to the DashboardTools.sd file in the dashboard service folder.
4. Under the Override Sharing Properties section, check box to share with your Organization or ERM group.
5. Set any folder options for where to store service.
6. Run tool.
7. Open Server Manager and verify Dashboard Tools geoprocessing tool exists and is running. Note the path for use in configuration.
8. Open Portal ad verify Dashboard item is shared correctly.
9. Apply Configuration

At this point you should have all the items needed to configure ERM components.

A PowerShell script is included that will use a csv to update configuration files for both Route Planner and ERM API.

1. From the extract ERM install package zip, open the deployment folder, and open the ERM\_Config\_Template.csv
2. Fill in the VALUE column for your environment values.
   1. The *ERM API Configuration* and *ERM Web Application* documentsavailable in the [ERM GitHub document repository](https://github.com/EsriPS/enterprise-route-management) has more detail on each value.
3. Save and close the csv.
4. Run the ERM\_UpdateConfig\_FindReplace.ps1 PowerShell script.
   1. The script assumes there is one csv at same level that has all values and that the overall folder structure of the release package has not been changed.
5. Script updates values in two files. You can validate files have been updated:
   1. <release package folder>\routeplanner\config.json
   2. <release package folder>\\middleware\src\config\config.js
6. ERM Middleware Server

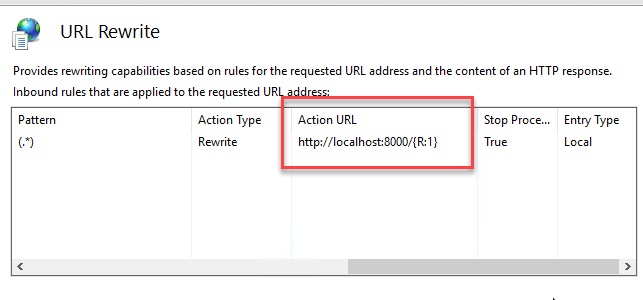
These steps will be performed on the machine hosting the ERM middleware API.

NOTE: for these instructions the deploy location is set as C:\arcgis\ERM. This can be changed to another drive or folder on the host machine if needed.

* 1. IIS & Forward Proxy

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

1. On the ERM middleware sever, open IIS
2. Create a new application under Default Web Site
   1. Alias = ermapi
   2. Use Default AppPool
   3. Set the 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 [localhost:8000](http://localhost:8000/)
      1. Leave out the http:// or it will get duplicated in the URL.
   6. After the rule is created, review the Action URL in the dialog.



|  |  |
| --- | --- |
|  | * When you create this rule, a web.config file will be created in the physical path location. If you delete this file during an application upgrade, will need to add the rule again |

* 1. Install ERM API Service

If this is not the initial install, review section 4.2 for update instructions.

1. Create a folder C:\arcgis\ERM
   1. This should match the physical location you added in IIS. If you went through those steps, folder will already be created.
2. From the extracted ERM release package zip, copy the Middleware and Scripts folder into C:\arcgis\ERM.
   1. This would be from the same location you ran the configuration script on.
3. Open a command prompt with run as Administrator option and navigate to C:\arcgis\ERM\scripts\middleware directory.
4. Run the following command:
   1. *node install-middleware-windows-service.js*
      1. Validate Install
5. Open the Windows Services and make sure that there is a service called 'ERM API'. If it's not started, start it. If the service does not exist, or you can't start it, check the log files located in the middleware\src\daemon directory.
6. In a browser window, navigate to http://localhost:8000/ and you should see "Welcome to the Enterprise Route Management API".
   1. If this does not work, verify that the Reverse Proxy is setup from the Environment Setup Guide.
7. Check that an ERM API log file is created in the location specified in the config file.
   1. If this does not work, verify that the path to your location is set correctly.
8. From a different machine that will be used to access the route planner app, navigate to http://<your middleware machine URL>/ermapi and you should see "Welcome to the Enterprise Route Management API".
   1. If this does not work, verify that middleware machine IIS has been configured correctly and is publicly (or intranet) available.

|  |  |
| --- | --- |
|  | * If you make any changes to the API config going forward, you need to restart the Windows service to apply the changes. |

* 1. ERM API Log

ERM has its own custom logging through the API. By default, the ERM services log to a “warn” level. ERM is designed to follow the same logging level convention as ArcGIS Server. Details on the levels can be found [here](https://enterprise.arcgis.com/en/server/latest/administer/windows/work-with-server-logs.htm#ESRI_SECTION2_6613A874BF944E28BBF0979DD4327670).

It is recommended to not use a level more detailed than “warn” in a daily production environment, unless needed to debug an issue. Setting the logging to a very detailed level can cause performance degradation for processes such as Create Plan.

Logging levels and details are set in the ERM API configuration file. There is a section for the API, and another section for the Workforce Sync service (if being used).

* Level
  + Defines the amount of detail included in log files
* Filename
  + Defines the log file path and name.
  + Use either / or \\ for the path syntax. A single \ will not read
* Maxsizeinbytes
  + Defines how large log file will get before a new one is made.
* Maxfiles
  + Defines how many log files will be kept. Once limit is reached the oldest is deleted.

// logging for ERM API

config.logging = {

level: process.env.LOG\_LEVEL || "warn", // silly, debug, verbose, info, warn, error

filename: process.env.LOG\_FILE || "C:\ERM\logs\erm\_api.log",

maxsizeinbytes: process.env.LOG\_SIZE\_BYTES || 20000000,

maxfiles: process.env.LOG\_MAX\_FILE\_COUNT || 15,

};

|  |  |
| --- | --- |
|  | * Logfile and folder will be created on service startup. If you delete the log file restarting the API service will recreate |

1. Route Planner

These steps will be performed on the Web Server hosting the Route Planner application.

* 1. Deployment

The ERM files for the Route Planner app need to be deployed to the web server and then the app configuration updated.

1. From the extracted ERM release package zip, copy the routeplanner folder into the wwwroot folder of your web server.
   1. This would be from the same location you ran the configuration script on.
2. In a browser window, navigate to <https://[your> URL or server name]/routeplanner and the web application should load.
   1. Browser Support

The Route Planner application is designed to be used with Chrome, Firefox, or Edge. **Internet Explorer is not supported.**

Route Planner is designed to use the default font size settings in the browser. Changing these settings can cause user interface and functionality issues.

1. Updating ERM

This section deals with making changes to an existing ERM deployment, including upgrading to a newer version.

* 1. Updating Feature Services

It is recommended to store your ERM data in either file or enterprise geodatabases, then make edits there and republish feature services rather than manage changes directly in the feature services. This way changes can be made in a single location and then published to multiple environments.

|  |  |
| --- | --- |
|  | * If the ERM\_Registry service is deleted or overwritten, existing plans will no longer be available in the ERM Route Planner application. |

You can overwrite existing feature services rather than deleting and publishing fresh. For Table layers you will need to delete the existing service and republish. If you use the Publish ERM Data tool, it will delete the existing service and re-publish with same name.

Manual steps for overwriting:

1. Open the ArcGIS Pro project.
2. In Pro, sign into the Portal that the services will be published to.
3. Open the map that needs to be republished.
4. Activate Share tab and select Web Layer > Overwrite Web Layer
5. A dialog with existing services will be shown.
6. Select the service to overwrite.
7. Share Web Layer panel will open with all settings filled in.
8. Adjust any settings as needed.
9. Analyze the service.
   1. Can ignore template warnings.
10. Publish service.
    * 1. Data Updates

Any data in ERM\_Plan\_Defaults (Depots, Routes, etc) can be updated by simply editing the local copy of data and then republishing service.

* + 1. Data Model Updates

New versions of ERM may include updates to the underlying Data Model. Release notes should call out the specific updates. Depending on the number of updates, these can be applied in several ways.

* Manually make changes to your local copy and republish.
* In your existing geodatabase
  + rename current feature class.
  + copy in feature class with updated schema.
  + Use core geoprocessing tools to load data from old feature class into new.
  + Delete old feature class.
  + Republish new.
  1. Updating Middleware
     1. Configuration

If only making configuration changes, such as updating a service path, the change can be made in the middleware/src/config/config.js file. Then the ERM API service needs to be restarted through the Windows Service console.

* + 1. Upgrade

For upgrading to a new version, you will need to use the uninstall scripts with the existing so that the build numbers match. The Node uninstall script expects the service it is uninstalling to have a specific description.

Once the Middleware is uninstalled and files are removed, the upgrade would follow the same steps as a clean install. See earlier section for proceedure.

Procedure for uninstall:

1. Stop the ERM API Service in the Windows Service console.
2. If you have the ERM Workforce Sync service also installed, stop that service as well.
3. Uninstall current middleware.
   1. Open an admin command prompt.
   2. Browse to C:\arcgis\ERM\scripts\middleware (or wherever this folder was deployed)
   3. Run command: ***node uninstall-middleware-windows-service.js***
   4. Close command prompt.
   5. Open Windows Service console and make sure ERM API service is not there.
4. Delete folders:
   1. C:\arcgis\ERM\scripts
   2. C:\arcgis\ERM\middleware
   3. Updating Route Planner

Any changes to configuration can be made in the web app config.json file for the deployed application. Users will need to clear their browser cache to pick up the changes. For some changes they may also need to clear local storage. Steps differ depending on browser being used, please refer to online help for your browser.

For deploying a new version, simply delete the existing routeplanner folder in the wwwroot folder and follow steps from initial install. Then have Users clear cache and local storage of their browser.

1. Readiness Checklist

Below is a checklist of items handled in the Environment Setup Guide that should be completed before application is deployed.

|  |  |  |
| --- | --- | --- |
| **#** | **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 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 |  |
| 12 | Custom Business System Integration is available (or plan to use sample BSI) |  |

1. App Deployment Checklist

Below is a checklist for deploying the ERM application.

|  |  |  |
| --- | --- | --- |
| **#** | **Task** | **Complete** |
| 1 | Create Web Map in Portal |  |
| 2 | Create Web Application in Portal |  |
| 3 | Create Portal Groups for locations |  |
| 4 | Create ERM Role in Portal |  |
| 5 | Assign users to groups and roles |  |
| 6 | Load my data into file geodatabases |  |
| 7 | Publish feature services |  |
| 8 | Update Middleware config file |  |
| 9 | Install Middleware API |  |
| 10 | Copy web application to web server |  |
| 11 | Update web configuration file |  |
| 12 | Run Sanity Tests |  |
|  |  |  |
|  |  |  |
| **#** | **Optional Tasks** | **Complete** |
| 1 | Create a template web map |  |
| 2 | Publish feature services with sample data |  |
| 3 | Publish sample BSI Service |  |

1. Web Map Template

You have the option to create a web map template that Route Planner will use when creating a plan. This will allow the ERM administrator to have extra layers in the plan, such as weather or traffic, along with being able to simplify updating symbology and labeling once ERM is deployed. This should be a separate web map than the empty one created in section 3.1.1.

If you do not define a template web map, a new plan is created with just the basic layers from the ERM\_Plan\_Template feature service.

This is a task that needs to be completed once ERM is completely deployed and able to create plans. If you are using the Dashboard functionality, some of this setup is duplicate with setting up a template Dashboard.

1. Log into ERM application as user that will be owner of the ERM items.
2. Create a plan.
3. Remove the plan from the Registry table so the app will no longer use.
   1. Open Pro and log into Portal as admin user that owns the ERM\_Registry feature service.
   2. Add the Registry table to a map
   3. Find entry for newly created plan and delete record. Save edits.
4. Log into Portal as same user that created the plan.
5. Find the Web Map and Feature Layer for the new plan
6. Mark item as not to delete
   1. Open item to details page
   2. Click Settings
   3. Under Delete Protection section, check the “Prevent this item from being accidentally delete” option.
   4. Repeat for Web Map and Feature Layer
7. Rename Web Map to “ERM Template Map” or another identifying name.
8. Rename Feature Layer to “ERM Template Features” or another identifying name.
9. If you have a folder to store all ERM items, move the map and feature layer into it.
   * + 1. Configure Template Web Map
10. Open your template web map through Portal.
11. Leave all existing ERM layers in the map. You can make following adjustments:
    1. Visibility
    2. Symbology
       1. Do not change for GeoOrders, that is controlled by ERM. You can set Cluster Points.
       2. Do not change for Routes, can be adjusted in ERM API configuration.
    3. Order/Sequence of Layers
       1. Note that in the web config you must set index values for layers. Those values come from the feature service and not the template web map.
    4. Labeling
    5. Add Popups
       1. Except for GeoOrder and Route layers. They get their popup information from the ERM web app configuration.
    6. Cluster Points
       1. This can be helpful to set on GeoOrders layer.
       2. To have the counts label in the cluster circle, will need to configure in the newer version of Web Map viewer.
       3. Clusters will need to be configured to have popups for user to be able to click on a Cluster point and get list of underlying orders
12. Add additional layers as needed. These could be:
    1. Extra ERM layers such as MDM Locations or GPS Truck Locations
    2. Living atlas layers such weather
    3. Traffic feeds
    4. Industry specific layers your organization hosts.

**NOTE:** If you will be using Living Atlas layers in your map, there is a known bug with the Python API. There is a work around that involves updating python files on the server. If you will be using these types of layers, the ERM project team can help with implementation.

Use the ERM\_Plan\_Template map for display settings of the layers.

* For all the layers except Routes, set the symbology for how they will display in the ERM application.
  + If you will be using a template web map, can skip this step
  + The default symbology for Routes is generated when Plan is created. The symbol shown in map is not used.
* Field order and visibility will be set in the web application configuration file.

1. Create Dashboard Template

ERM must be deployed before you can create data for a dashboard template. You can use a plan created with sample BSI to create a template.

1. Log into ERM application as user that will be owner of the dashboard items.
2. Create a plan.
3. Can optionally Solve/Commit to change Routes and Orders. Having different values set can help with configuration of dashboard.
4. Remove the plan from the Registry table so the app will no longer use it.
   1. Open Pro and log into Portal as admin user that owns the ERM\_Registry feature service.
   2. Add the Registry table to a map.
   3. Find entry for newly created plan and delete record. Save edits.
5. Log into Portal as same user that created the plan.
6. Find the Web Map and Feature Layer for the new plan.
7. Mark item as not to delete.
   1. Open the item to Details page.
   2. Click Settings
   3. Under Delete Protection section, check the “Prevent this item from being accidentally deleted” option.
   4. Repeat for both Web Map and Feature Layer
8. Rename Web Map to “ERM Dashboard Template Map” or another identifying name.
9. Rename Feature Layer to “ERM Dashboard Template Features” or another identifying name.
10. Optional: Create an “ERM Dashboard” folder and move the map and feature layer into it.

Add Additional Data Layers

You can optionally add additional layers that are not in the ERM Template service and included in the plan web map. These could be truck locations, weather, or any other layers that would be helpful in the dashboard.

**NOTE**: All additional layers to be added into the dashboard will also be in the plan map that the Route Planner users will see. The [*ERM Application Deployment Guide*](https://github.com/EsriPS/enterprise-route-management/blob/master/Install-Deployment/ERM%20Application%20Deployment%20Guide.docx) has more information on using a web map template for Create Plan.

If you do not want additional layers, skip to the next section.

1. From Portal, open your ERM Dashboard Template Map.
2. Add additional layers into your map. Save changes.
3. Record the ID of the map.
4. In the ERM API config.js file, set the config.templateWebMapId value to your map ID.
5. Restart the ERM service.
6. Make sure the map is shared with the organization or any overriding ERM group.

**NOTE:** If you will be using Living Atlas layers in your map, there is a known bug with the Python API. There is a work around that involves updating python files on the server. If you will be using these types of layers, the ERM project team can help with implementation.

Create Template Dashboard

Dashboard can be configured to display any available information from the plan. Resources for available elements and how to configure are available through Esri online help [here](https://www.esri.com/en-us/arcgis/products/arcgis-dashboards/resources).

1. In Portal, in the ERM Dashboard folder, click Create and choose Dashboards.
   1. Name = “ERM Dashboard”
      1. Note that this name will be the prefix for all created dashboards, i.e. “ERM Dashboard – Plan GOL 20200831”
   2. Add tags and summary.
2. Add elements as needed. Point all to use the web map created in step 2.1

Update Configuration

Main configuration is within the ERM Middleware API configuration file. There is also a value in the web app config to show the Dashboard buttons.

API Config

1. On the machine where ERM Middleware API is deployed open the config file
   1. Located at <install location>\middleware\src\config.json
2. Open the config file
3. Set the dashboardTemplateItemId value to the ID of the template dashboard created previously.
4. If using extra data layers in your dashboard, set the config.templateWebMapId to the ID of the map used to make the dashboard.
   1. If not using extra data layers, leave values set to “”
5. Update geoprocessing path as needed. If you left folder as default during publishing should not need to change.
6. Restart the ERM API service after making any changes.

config.dashboardTemplateItemId = "d8d4a3f9d413418b91a553c8e73f18a1";

config.dashboardGPUrl =

config.hostingServerBaseUrl +

"/rest/services/GP/DashboardTools/GPServer/Create%20Plan%20Dashboard";

Web Config

There is a configuration value to show or hide the Dashboard buttons. Need to verify that it is set to true, so buttons are available.

1. On the web hosting machine, open the app configuration file.
   1. Located under routeplanner\config.json
2. Verify that showDashboardControls is set to true.
3. Base Functionality Test

After all deployment steps are complete, run the following procedures to test that all base functions are configured and operating correctly.

1. Log into Route Planner app.
2. Click Create New Plan.
   1. Verify correct list of locations are available to create.
   2. Choose a date and time that will pull in orders for chosen date. Otherwise the Optimization will fail. If using the test BSI service, this would be January 1st of current year at 12:00.
3. New Plan created and user is taken to Edit Plan page.
   1. May need to refresh application for all items to show up for very first plan.
4. Open Routes tab.
5. Click Run Route Optimization.
6. Routes are solved and geometries assigned.
7. Document History

| Date | Description | Editor |
| --- | --- | --- |
| 2/3/2020 | Initial Draft v0.1 | Mike Nelson |
| 4/1/2020 | Various updates | Mike Nelson |
| 6/12/2020 | Various updates | Mike Nelson |
| 7/2/2020 | Apply feedback from customer development deployment | Mike Nelson |
| 9/10/2020 | Updates for how Routes are symbolized | Mike Nelson |
| 9/25/2020 | Misc. updates focused on redeployment | Mike Nelson |
| 9/30/2020 | GPS/GeoEvent update | Mike Nelson |
| 11/3/2020 | Note about browser font and zooming | Mike Nelson |
| 3/1/2021 | Removed step to use “user can update all items” in Group settings | Mike Nelson |
| 4/30/2021 | Added info for using web map for Create Plan. Plus, other minor updates | Mike Nelson |
| 7/12/2021 | Updates for Support 3 Release   * New Barrier layer * Updates to procedure for publishing feature services | Mike Nelson |
| 8/18/2021 | Updates for Support 4 Release   * More detail on configuring and logging settings | Mike Nelson |
| 3/18/2022 | General updates and procedure refinements | Mike Nelson |
|  | Updates for 4.1 release   * Note on configuring zones to a route | Mike Nelson |