Last Update: August 19, 2022

Enterprise Route Management

*Application Deployment Guide*

**Table of Contents**

**Section Title Page**

[1.0 Overview 3](#_Toc111891291)

[1.1 System Prerequisites 3](#_Toc111891292)

[1.2 ERM Install Package 3](#_Toc111891293)

[2.0 Enterprise 5](#_Toc111891294)

[2.1 Portal Items 5](#_Toc111891295)

[2.1.1 Web Map for App 5](#_Toc111891296)

[2.1.2 Application Item 6](#_Toc111891297)

[2.1.3 Create Location Groups 7](#_Toc111891298)

[2.1.4 Users 7](#_Toc111891299)

[3.0 ERM Feature Services 8](#_Toc111891300)

[3.1 Load Customer Data 8](#_Toc111891301)

[3.1.1 ERM\_Plan\_Defaults 9](#_Toc111891302)

[3.1.2 ERM\_Plan\_Template 9](#_Toc111891303)

[3.1.3 ERM\_Registry 9](#_Toc111891304)

[3.1.4 MDM\_Locations 9](#_Toc111891305)

[3.1.5 GPS\_Template 10](#_Toc111891306)

[3.2 Travel Modes/ERM\_Solve\_Parameters 10](#_Toc111891307)

[3.3 Publish Feature Services 12](#_Toc111891308)

[3.3.1 Batch Tool 12](#_Toc111891309)

[3.3.2 Manual Publish 13](#_Toc111891310)

[4.0 ERM Middleware Server 14](#_Toc111891311)

[4.1 IIS & Forward Proxy 14](#_Toc111891312)

[4.2 Install ERM API Service 15](#_Toc111891313)

[4.2.1 Validate Install 15](#_Toc111891314)

[4.3 ERM API Log 16](#_Toc111891315)

[5.0 Web Sever 17](#_Toc111891316)

[5.1 Deployment 17](#_Toc111891317)

[5.2 Browser Support 17](#_Toc111891318)

[6.0 Business System Integration 18](#_Toc111891319)

[6.1 Sample BSI Service 18](#_Toc111891320)

[6.1.1 Setup BSI Files 18](#_Toc111891321)

[6.1.2 Publish BSI Service 18](#_Toc111891322)

[6.2 Sample Data 19](#_Toc111891323)

[7.0 Updating ERM 20](#_Toc111891324)

[7.1 Updating Feature Services 20](#_Toc111891325)

[7.1.1 Data Updates 20](#_Toc111891326)

[7.1.2 Data Model Updates 21](#_Toc111891327)

[7.2 Updating Middleware 21](#_Toc111891328)

[7.2.1 Configuration 21](#_Toc111891329)

[7.2.2 Upgrade 21](#_Toc111891330)

[7.3 Updating Route Planner 22](#_Toc111891331)

[Appendix A Readiness Checklist 23](#_Toc111891332)

[Appendix B App Deployment Checklist 24](#_Toc111891333)

[Appendix C Web Map Template 25](#_Toc111891334)

[Appendix D Base Functionality Test 27](#_Toc111891335)

[Appendix E Document History 29](#_Toc111891336)

1. Overview

This Application Deployment Guide serves to document steps for deploying the components for the Enterprise Route Management application. The first sections cover an initial or clean install. See [Section 7](#Section7) for instructions and note 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 of EMR environment should already be setup. Depending on 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).
  + Includes Street Map Premium install for Network services
* Server to host ERM middleware API
* Web Server to host ERM Route Planner Application
* Ancillary items installed or available, such as Node.js install, 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
* Routeplanner – web application
* Middleware – code files for ERM API and Workforce Sync services
* Scripts – code for publishing ERM API and Workforce Sync services
* Services – contains ArcGIS Pro project, empty file geodatabases to load customer data into, and tools to publish ERM Feature Services.
* Integration service – file to publish a sample BSI GP service.
  + Use for testing and system validation. Only needed if not using customer BSI.
* Services\_Sample – contains ArcGIS Pro project and sample data to publish to use with sample Integration services
  + Only needed if using sample BSI.

1. Enterprise

This section covers ERM items to be published to 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 just 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.

For organizational purposes, it is recommended these items are placed in a folder in Portal called ERM Items, or something similar, for easy reference.

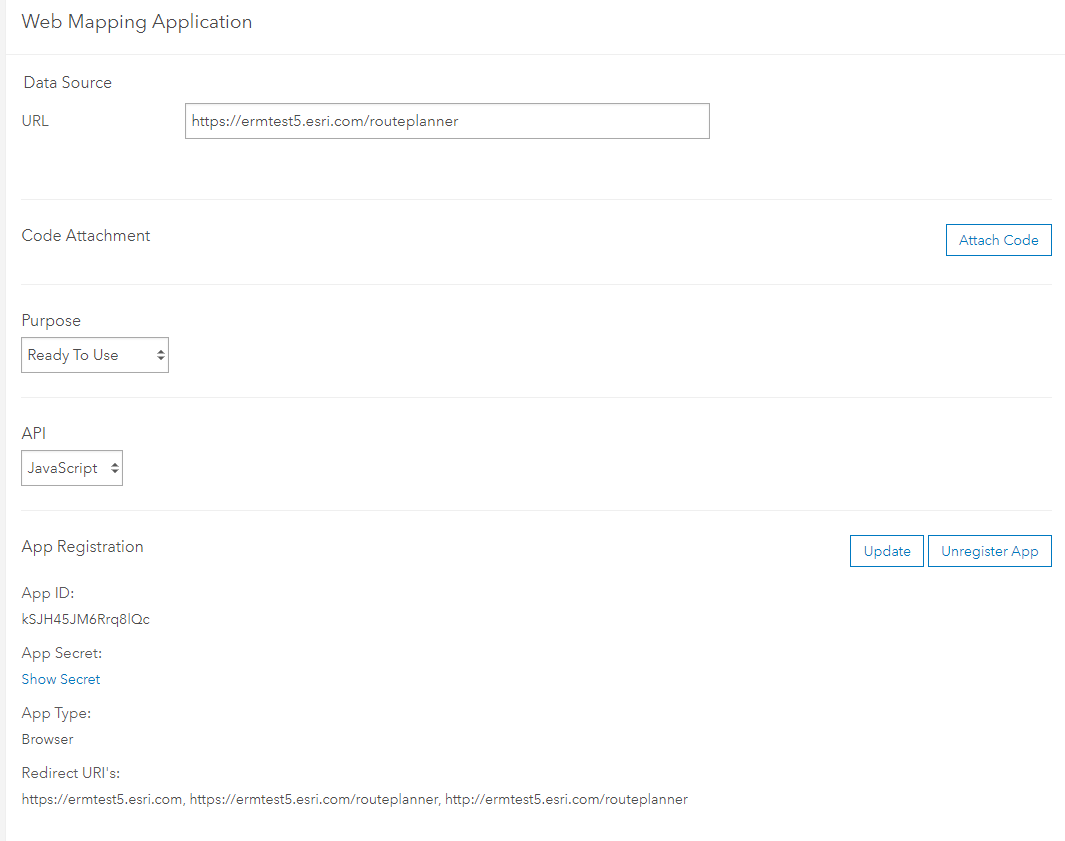
* + 1. Web Map for App

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 name such as “ERM default map”. Name not used by app
3. Do not add any layers
4. Can leave Basemap as default, application configuration will set that
   1. Note that the very first time Route Planner is loaded, before any plans are created for a location, user will see this basemap. So you could set to the same basemap that the application configuration will use
5. Initial extent does not matter, application will define
6. 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)
7. Open Settings of map and mark “Prevent item from being accidentally deleted”
   * 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 just put you in details)
9. Share the application item with your Organization or a defined ERM group
10. Open Settings and 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. Create Location Groups

Create a Group for each Dispatch Location.

1. Create group with the name of location
2. To each Group add a tag of the format "dispatch-location-xxx"(where xxx denotes the dispatch location. For example, "dispatch-location-OCC", "dispatch-location-COV", etc.)
3. Set that People in organization can view group
4. Set that Group Members can contribute content
5. Set that group members can update only their items
6. Add users to group

A ERM\_CreateGroups.py script is available in the [GitHub document repository](https://github.com/EsriPS/enterprise-route-management) in the Operational-Documentation folder. This can be adjusted to create groups for all your locations.

* Update the group\_list variable with list of location names
* Update the variables holding Portal credentials
* Can optionally add tags to tag\_list variable
  + 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.
* User must be member to 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. 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 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. Instructions for publishing are in the Business System Integration section

* 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
   1. May receive a message that project was created with a newer version. If on Pro 2.6 or later, should work correctly
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. ERM\_Solve\_Parameters
   5. MDM\_Locations
   6. 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).
* 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 section 4.2.6 Configure Travel Modes/ERM\_Solve\_Parameters
  + 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. Optional functionality that can be configured, especially if using Workforce in conjunction with Route Planner.

* 1. Travel Modes/ERM\_Solve\_Parameters

It is recommended that you use Travel Modes that are configured on your Routing Service. These replace the ERM\_Solve\_Parameters tables used by earlier versions of ERM. If using Travel Modes, the ERM\_Solve\_Parameters tables can be left empty.

**NOTE: ERM\_Solve\_Parameters tables and supports are planned for removal at release 4.0**

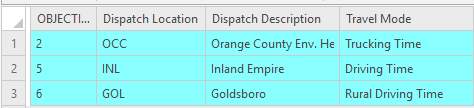
* + - 1. Configure Travel Modes

In Portal, under Organization > 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. Configure ERM\_Solve\_Parameters (Depreciated)

**NOTE: ERM\_Solve\_Parameters tables and supports are planned for removal at release 4.0**

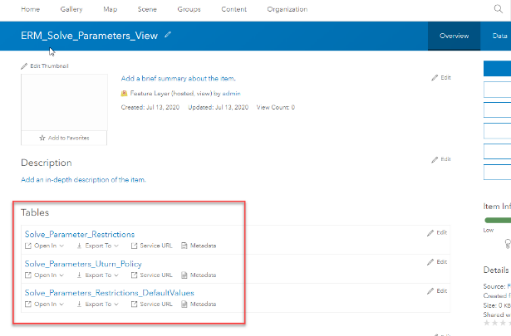
The ERM\_Solve\_Parameters table represent initial ERM functionality, where all travel rules were stored in a separate service. Only need to do this step if not using Travel Modes.

1. Open ERM\_Solve\_Parameters map
   1. The Solve\_Parameters\_Restrictions\_DefaultValues table contains all the solve parameter values used by the VRP.
      1. Update the Restriction Parameter Value for each parameter to match how all or most locations should be.
   2. Update the Solve\_Parameter\_Restrictions table for each of your locations.
      1. Use the Append tool to load all rows from Solve\_Parameters\_Restrictions\_DefaultValues table into the Solve\_Parameter\_Restrictions table.
      2. Calculate the Dispatch Location name for all rows just loaded in to your first Location.
      3. Repeat the append and calculate name process for each location. If there are many locations, it may be better to script the process.
      4. If an individual or subset of locations need a solve parameter changed, it would be done in the Solve\_Parameter\_Restrictions table.
      5. A ERM\_CopySolveParameters.py script is available in the [GitHub document repository](https://github.com/EsriPS/enterprise-route-management/tree/master/Operational-Documentation) to batch copy parameters for each depot (steps i-iv)
   3. Update the Solve\_Parameter\_Uturn\_Policy table for each of your locations.
      1. Add a row for each location. Set Dispatch Location to name of location.
      2. Set the U-turn policy for each location.
2. Open the ERM\_Plan\_Defaults map
   1. In the DispatchLocation table, set Travel Mode = Custom for any locations that will get their travel rules from the ERM\_Solve\_Parameters service.
   2. 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. Assumes you are using the ERM\_services project you are in and that data lives in fgdbs folder.
   2. Maps with Data to Publish – can choose all or any subset.
   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 end of service name.
      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 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. ERM\_Solve\_Parameters table
   5. ERM\_Solve\_Parameters\_View – view created from ERM\_Solve\_Parameters table
   6. MDM\_Locations feature layer
   7. GPS\_Template feature layer

|  |  |
| --- | --- |
|  | * If Python API is version is before 1.8, the View may not get published correctly. To validate, open the view and verify there are 3 tables listed. If not, can delete the view and recreate using the steps found in the “Manual Publish” section |



* + 1. Manual Publish

If there are tool issues, or you just want to manually run through same steps, these are

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. Set Summary = Name and add tag ERM
   4. Select "Copy all data" option in the Data section.
   5. Set "Feature" option in the Layer Type section.
   6. Share with Organization (Except for ERM Solve Parameters)
   7. Select Configuration tab and click pencil icon for Feature layer
   8. Check Enabled Editing and Export Data options
   9. Analyze the service.
      1. Ignore template warnings
   10. Publish service.
4. Make a View for ERM Solve Parameters
   1. Log into Portal
   2. Open details for ERM\_Solve\_Parameters
   3. Click Create View
   4. Name ERM\_Solve\_Parameters\_View
   5. Share view with Organization
5. ERM Middleware Server

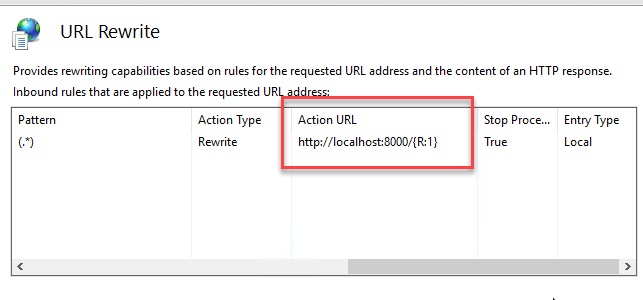
These steps will be performed on the machine hosting the ERM middleware API.

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. 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 [localhost:8000](http://localhost:8000/)
      1. Leave out the http:// or it will get duplicated in the URL.
   6. After 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
2. Create a Logs folder C:\arcgis\ERM\Logs
3. From the extracted delivery zip, copy the Middleware and Scripts folder into C:\arcgis\ERM.
4. Open the middleware/src/config/config.js file
   1. Update the 3 URLs for your Portal, Server and ERM API.
   2. If you have published your feature services with different names, will need to update the 4 feature server layer paths with new names.
   3. In the logging section, add the log folder directory in front of the log file name. Make sure to use forward slashes or double backslash.
      1. See section 8.0 for details on level of logging to capture.
5. Open a command prompt with run as Administrator option and navigate to the middleware directory (C:\arcgis\ERM\middleware).
6. Run the following commands:
   1. *npm install* 
      1. Do not need to run this if folder already has node\_modules folder. This can be provided by Esri if there are permissions issues with node install
   2. *cd ../scripts*
   3. *npm install*
      1. Do not need to run this if folder already has node\_modules folder. This can be provided by Esri if there are permissions issues with node install
   4. *node install-middleware-windows-service.js*
      1. Validate Install
7. 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.
8. 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.
9. 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.
10. 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.
    2. 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,

};

1. Web Sever

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. Extract the ERM Web Application zip file.
2. Copy the routeplanner folder into the wwwroot folder of your web server.
3. Under root routeplanner folder, open the config.json file.
4. Update items for your environment:
   1. portalUrl
   2. apiUrl
   3. depotLocationLayerUrl
   4. PlanRegistryUrl
   5. appId
   6. appItemId
   7. webmapId
5. 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. Business System Integration

A Business System Integration (BSI) service needs to be published that links between 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.

* 1. Sample BSI Service

A sample service can be setup to simulate a BSI service. These instructions assume the host server machine is a windows machine and Pro is installed.

|  |  |
| --- | --- |
|  | * 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 host machine, create the following folders:
   1. C:\arcgis\ERM\BSI
   2. C:\arcgis\ERM\Commit
2. From the extracted delivery, copy the jsonIntegration folder (found under integration service\data) into C:\arcgis\ERM\BSI
3. From the extracted delivery, copy the integration service folder onto machine with Pro. Should contain data and python folders.
4. In the python folder, open config folder
5. Update the file path in both the GetPlanObjects.ini and JSONGetPlanObjects.ini to where you copied sample json (C:\arcgis\ERM\BSI\ jsonIntegration)
6. Update the file path in CommitPlan.ini to match the created Commit folder (C:\arcgis\ERM\Commit)
7. From the python folder, open the Publish\_BSI\_TestService.py in a python editor (such as IDLE).
8. Update the Portal credential variables for your environment.
9. Can optionally update the bsi\_layer value for what the service will be called. This name would need to be updated in the Middleware config file.
   * 1. Publish BSI Service
10. Open ArcGIS Pro and log into your Portal.
11. Open the Upload Service Definition tool from the geoprocessing pane.
12. Point to the BSI.sd file in the integration service\python folder.
13. Under the Override Sharing Properties section, check box to share with your Organization or ERM group.
14. Set any folder options for where to store service.
15. Run tool.
16. Open Server Manager.
17. 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
18. 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
19. From the extracted python folder, copy the config folder into the p20 folder.
20. Can open Server Manager and verify item is published and is started.
21. Stop and restart the BSI service to verify it picks up the configuration update.
    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. 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 existing service and republish 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. 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 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 same steps as a clean install. See earlier section for steps.

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 (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 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 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 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 |  |
| 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.
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. 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 Optimization will fail. If using the test BSI service, this would be 1/1/2021 12:00.
3. New Plan created and user is taken to Edit Plan page.
   1. May need to refresh app 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.

1. 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 |