Last Update: June 16, 2025

Enterprise Route Management

*Deployment Guide*

**Table of Contents**

**Section Title Page**

[1.0 Overview 3](#_Toc201142603)

[1.1 System Prerequisites 3](#_Toc201142604)

[1.2 ERM Install Package 3](#_Toc201142605)

[2.0 Enterprise 5](#_Toc201142606)

[2.1 Portal Items 5](#_Toc201142607)

[2.1.1 Web Map for App 5](#_Toc201142608)

[2.1.2 Application Item 6](#_Toc201142609)

[2.1.3 Create Location Groups 6](#_Toc201142610)

[2.1.4 Users 7](#_Toc201142611)

[3.0 ERM Feature Services 8](#_Toc201142612)

[3.1 Load Customer Data 8](#_Toc201142613)

[3.1.1 ERM\_Plan\_Defaults 8](#_Toc201142614)

[3.1.2 ERM\_Plan\_Template 9](#_Toc201142615)

[3.1.3 ERM\_Registry 9](#_Toc201142616)

[3.1.4 MDM\_Locations 9](#_Toc201142617)

[3.1.5 GPS\_Template 10](#_Toc201142618)

[3.2 Travel Modes 10](#_Toc201142619)

[3.2.1 Configure Travel Modes 10](#_Toc201142620)

[3.3 Publish Feature Services 11](#_Toc201142621)

[4.0 Business System Integration 12](#_Toc201142622)

[4.1 Esri BSI Service 12](#_Toc201142623)

[4.1.1 System Setup 12](#_Toc201142624)

[4.1.2 Setup BSI Files 13](#_Toc201142625)

[4.1.3 Publish BSI Service 13](#_Toc201142626)

[4.2 Configure ERM API 14](#_Toc201142627)

[5.0 Dashboard Service 15](#_Toc201142628)

[5.1.1 Publish Dashboard Service 15](#_Toc201142629)

[6.0 Apply Configuration 16](#_Toc201142630)

[7.0 ERM Middleware Server 17](#_Toc201142631)

[7.1 Install ERM API Service 17](#_Toc201142632)

[7.1.1 Validate Install 17](#_Toc201142633)

[7.2 ERM API Log 18](#_Toc201142634)

[8.0 Route Planner 19](#_Toc201142635)

[8.1 Deployment 19](#_Toc201142636)

[8.2 Browser Support 19](#_Toc201142637)

[9.0 Updating ERM 20](#_Toc201142638)

[9.1 Updating Feature Services 20](#_Toc201142639)

[9.1.1 Data Updates 20](#_Toc201142640)

[9.1.2 Data Model Updates 21](#_Toc201142641)

[9.2 Updating Middleware 21](#_Toc201142642)

[9.2.1 Configuration 21](#_Toc201142643)

[9.2.2 Upgrade 21](#_Toc201142644)

[9.3 Updating Route Planner 22](#_Toc201142645)

[Appendix A Readiness Checklist 23](#_Toc201142646)

[Appendix B App Deployment Checklist 24](#_Toc201142647)

[Appendix C Base Functionality Test 25](#_Toc201142648)

[Appendix D Web Map Template 26](#_Toc201142649)

[Appendix E Dashboard Template 28](#_Toc201142650)

[Appendix F Using Sample Data 31](#_Toc201142651)

1. Overview

This Application Deployment Guide serves to document steps for deploying the components for the Enterprise Route Management (ERM) system. The ERM system is made up of several components:

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

|  |  |
| --- | --- |
|  | If you are installing a new version of ERM, see [Updating ERM](#UpdatingERM) section for steps on uninstalling existing items before installing update |

* 1. System Prerequisites

The following components for the EMR environment should already be set up. Depending on the environment there may be 3 separate machines, or some components might be on the same machine.

* Base deployment of ArcGIS Enterprise (Portal, Server and relational Data Store).
  + Including 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 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 [Readiness Checklist Appendix](#ReadinessChecklist) for list of prerequisites.

* 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.

Extract the release package 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 with the organization, all users will have access to these items.

Recommendations:

* Create items using the same administration account that will be used to publish services later.
* Can also add additional tags such as “ERM” to be able to easily search for all items used by ERM. Only required tags are included in instructions.
* Place all ERM items in a folder in Portal called ERM Items, or something similar, for easy access.

|  |  |
| --- | --- |
|  | The IDs/URLs of Portal items will be needed for configuration. As they are being created, you can store values in the ERM\_Config\_Template.csv that will be used in a later step to configure application and API. See [Apply Configuration](#ApplyConfiguration) section for details on file. |

* + 1. Web Map for App

The Route Planner application needs a blank web map to point at.

1. Open an empty web map.
2. Extent does not matter, application will define.
3. Do not add any layers. Leave the Basemap as default, application configuration will set what is used by Route Planner.
   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. Choose Save As.
5. Give a name such as “ERM default map”. Name is not used by app.
6. Share the web map with your organization or a defined ERM group.
7. Once map saved, open Settings of map and mark “Prevent item from being accidentally deleted.”
   1. It can optionally be marked as Authoritative as well.

|  |  |
| --- | --- |
|  | There is an optional step for creating a template web map that would define layer settings and contain non-ERM layers. See [Web Map Template Appendix section](#WebMapTemplate) for instructions. These should be two separate web maps. |

* + 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. Optionally set any Tags for organizing and add Summary.
7. Press Save to create the item.
8. Open item details.
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 = Other
13. Optionally add redirect URLs for your app with https and http.
    1. http://<your URL or machine name>/routeplanner
14. Set Application environment = Browser
15. Click Register App
16. Save the item ID found in the URL and the registered ID for the item.
17. Mark “Prevent item from being accidentally deleted.”
    1. It can optionally be marked 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 the group.
   3. Group Members can contribute content.
   4. Group members can update only their items.
      1. Users

Users that will be accessing the 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 the 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 Sample data included in the services\_Sample directory is to be used with the sample BSI. Directions for use 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 services folder, open the *ERM\_services.aprx* Pro project included with release
2. 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 the sequence that they are presented. The application expects the index value of layers to be a certain value. * These can be updated in the web app configuration file, but leaving in default order will save you from having to update those values with future application updates. |

* + 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](https://pro.arcgis.com/en/pro-app/latest/help/data/geodatabases/overview/import-data.htm) for more information on tools.

* GeoOrderTemplate – stores order locations if using the Esri BSI to read from ERM\_Plan\_Defaults (optional).
* 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).
* CollectionTemplate table – stores information on collections if using the Esri BSI to read from ERM\_Plan\_Defaults (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 the value.
* RouteTemplate table – add all available routes for all locations. If Routes have multiple specialties they should be separated by a space.
* SpecialtyNameTemplate table – enter information on available specialties (optional). Note that VRP does not support Specialty names with spaces.
* OrderPairTeamplate table – holds order pairs, if using order pairs and using the Esri BSI to read from ERM\_Plan\_Defaults (optional).
  + 1. ERM\_Plan\_Template

ERM\_Plan\_Template service is used to create the feature service for a plan. The ERM\_Plan\_Template map should not be adjusted, rather use a Template Web Map to alter symbology. See [Appendix C](#AppendixC) for instructions on creating a Template Web Map.

* + 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 dispatch location is serviced by several different yards. If not using this functionality, the 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 Dispatch Location name to relate the location to a central Dispatch Location.
   * 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 if you are tracking vehicle locations.

* 1. Travel Modes

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

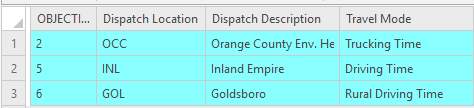
* + 1. Configure Travel Modes

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. See [online help](https://enterprise.arcgis.com/en/portal/latest/administer/windows/travel-modes.htm) for more information on Travel Modes.

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

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

|  |  |
| --- | --- |
|  | These steps assume you have loaded your data into empty ERM\_Plan\_Defaults and publishing those services. See [Using Sample Data](#UsingSampleData) section if wanting to publish sample data rather than your own. |

1. Open the ERM\_services Pro project included with release that was used to populate the data.
2. Connect to Portal as an administrator 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, you 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.
   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. Verify the items are shared with your organization or a defined ERM group.
8. 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. There are three options for the BSI:

* A customer developed and managed custom BSI service that integrates with a separate business system.
* Esri developed BSI service configured to read from ERM\_Plan\_Defaults feature service.
  + This configuration has limited functionality within ERM.
  + Data can be edited directly, or a separate integration service could populate this from a business system.
* Esri developed BSI service configured to read from JSON files.
  + This configuration can be used with sample data provided by Esri or with files created by customer from data in a business system.

It is expected that Esri will work with customers on which configuration will meet their requirements. Details on custom BSI or separate integrations are handled outside of this document.

This section deals with publishing and configuration of Esri developed BSI. There are additional steps when configuring the ERM Middleware service, covered later in this doc.

* 1. Esri BSI Service

If using the Esri BSI, use these instructions.

* + 1. System Setup

1. On the Enterprise machine, create the following folders:
   1. C:\arcgis\ERM\Commit
   2. C:\arcgis\ERM\BSI

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

* + 1. Setup BSI Files

|  |  |
| --- | --- |
|  | If using the Esri BSI to read from ERM\_Plan\_Defaults feature service, you do not need to setup these JSON files. Skip step 3. |

1. If using JSON files:
   1. For the Esri sample data JSONs, copy the jsonIntegration folder (found under integration service\data) into C:\arcgis\ERM\BSI on the Enterprise server.
   2. If generating your own JSON, place them in the C:\arcgis\ERM\BSI\jsonIntegration folder on the Enterprise server. Or point any tool that outputs JSON to this folder.
   3. In the python\config folder, update the file path in JSONGetPlanObjects.ini to where you copied sample json (C:\arcgis\ERM\BSI\ jsonIntegration).
2. If using Hosted Feature Service
   1. Load your data into ERM\_Plan\_Defaults
3. In the python\config folder, update the file path in CommitPlan.ini to match the created Commit folder (C:\arcgis\ERM\Commit)
   * 1. Publish BSI Service
4. Open ArcGIS Pro and log into your Portal.
5. Open the Upload Service Definition tool from the geoprocessing pane.
6. Point to the BSI.sd file in the integration service\python folder.
7. Under the Override Sharing Properties section, check box to share with your Organization or ERM group.
8. Set any folder options for where to store service.
9. Run tool.
10. Open Server Manager.
11. 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
12. 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
    1. If there is ap30 folder (depending on Enterprise version), put the config folder in the cd folder one level up.
13. From the extracted python folder, copy the config folder into the p20 folder.
    1. This is the folder where you updated the .ini files.
14. Open Server Manager and verify item is published and is started.
15. Open Portal and verify the item is shared correctly.
    1. Configure ERM API

Once the ERM API is deployed, there are two configuration values that deal with the BSI:

* config.bsiGPBaseUrl
  + This is set to the URL of the BSI service, whether using a custom service or Esri supplied.
* config.bsiGetPlanUrl
  + Endpoint called by Create Plan
    - If using Esri BSI with JSON, this will be set to JSON%20Get%20Plan%20Objects
    - If using Esri BSI with Feature Service, this will be set to Get%20Plan%20Objects
    - If using a custom BSI, set to the service endpoint that was created.

1. 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, the Dashboard items can be left 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.

See [Dashboard Template Appendix](#DashboardTemplate) for instructions on creating a dashboard template.

* + 1. Publish Dashboard Service

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 Middleware and Route Planner. A PowerShell script is included in release package 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 after running the script:
   1. <release package folder>\routeplanner\config.json
   2. <release package folder>\\middleware\src\config\config.js
6. ERM Middleware Server

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

* 1. Install ERM API Service

1. On the ERM API server, 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 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 set up 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 ERM API server IIS has been configured correctly and is publicly (or internally) 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 managing 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. 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 procedure.

Procedure for uninstall:

1. On the Middleware server, stop the ERM API Service in the Windows Service console.
2. 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. Open Windows Service console and make sure ERM API service is not there.
3. Delete folders:
   1. C:\arcgis\ERM\scripts
   2. C:\arcgis\ERM\middleware
   3. Updating Route Planner

See the Release Notes for a release to check for any configuration updates before updating an existing Route Planner deployment.

1. Make a copy of your existing config\_data.json file.
2. Delete the existing routeplanner folder on your web server.
3. Follow normal steps to deploy Route Planner.
4. If there were updates in release related to config\_data.json, make those updates to your original copy.
5. Once deployed, replace the config\_data.json with your original copy.
6. 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 exposed with URL that app can hit |  |
| 9 | Web Server configured |  |
| 10 | URL available for Route Planner app (if not using web server name in URL) |  |
| 11 | All certificates configured and available |  |
| 12 | Custom Business System Integration is available (or plan to use Esri BSI) |  |

1. App Deployment Checklist

Below is a checklist for deploying the ERM application.

|  |  |  |
| --- | --- | --- |
| **#** | **Task** | **Complete** |
| 1 | Create Web Map and Application in Portal |  |
| 2 | Create Portal Groups for locations |  |
| 3 | Assign users to groups and roles |  |
| 4 | Load customer data into ERM\_Plan\_Defaults  Or  Use services\_Sample |  |
| 5 | Publish feature services |  |
| 6 | Update Middleware config file |  |
| 7 | Install Middleware API |  |
| 8 | Copy web application to web server |  |
| 9 | Update Route Planner configuration file |  |
| 10 | Run Base Functionality Tests (Appendix E) |  |
| 11 |  |  |
| 12 |  |  |
|  |  |  |
|  |  |  |
| **#** | **Optional Tasks** | **Complete** |
| 1 | Create a template web map |  |

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 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. 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 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. 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. Using Sample Data

Esri provides a few sample locations with sample orders and routes. This includes sample data to publish and corresponding JSON data that the Esri BSI will use for Create Plan/Refresh.

**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. Names assume no suffix was entered.
   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

**BSI with Sample Data**

See the [Business System Integration](#BSI) section for details on setting up Esri BSI service to use the sample JSON files.