September 2020

Enterprise Route Management

*Application Deployment Guide*

**Table of Contents**

**Section Title Page**

[1.0 About This Document 3](#_Toc51076132)

[1.1 Purpose 3](#_Toc51076133)

[1.2 Authors & Contributors 3](#_Toc51076134)

[1.3 Revision History 3](#_Toc51076135)

[2.0 Application Overview 4](#_Toc51076136)

[2.1 System Prerequisites 4](#_Toc51076137)

[2.2 File Prerequisites 4](#_Toc51076138)

[3.0 ArcGIS Enterprise 5](#_Toc51076139)

[3.1 Portal Items 5](#_Toc51076140)

[3.1.1 Web Map for App 5](#_Toc51076141)

[3.1.2 Application Item 5](#_Toc51076142)

[3.1.3 Create Location Groups 5](#_Toc51076143)

[3.1.4 Users 6](#_Toc51076144)

[4.0 ERM Feature Services 7](#_Toc51076145)

[4.1 Sample Data 7](#_Toc51076146)

[4.2 Load Customer Data 7](#_Toc51076147)

[4.3 Batch Publish 9](#_Toc51076148)

[4.4 Manual Publish 10](#_Toc51076149)

[4.4.1 Updating - Overwrite Existing 10](#_Toc51076150)

[5.0 ERM Middleware 12](#_Toc51076151)

[5.1 Initial Install 12](#_Toc51076152)

[5.2 Updating the Middleware 12](#_Toc51076153)

[6.0 Web Application 14](#_Toc51076154)

[6.1 Initial Install 14](#_Toc51076155)

[6.2 Updating the Web Application 14](#_Toc51076156)

[7.0 Business System Integration 15](#_Toc51076157)

[7.1 Sample BSI Service 15](#_Toc51076158)

[8.0 Sanity Test 16](#_Toc51076159)

[Appendix A Readiness Checklist 17](#_Toc51076160)

[Appendix B App Deployment Checklist 18](#_Toc51076161)

1. About This Document
   1. Purpose

This Application Deployment Guide serves to document steps for deploy the Enterprise Route Management application.

There are precursor steps for setting up the underlying environment. These steps are included in separate document *ERM Environment Setup Guide.*

* 1. Authors & Contributors

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

* 1. Revision History

|  |  |  |
| --- | --- | --- |
| Date | 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 |

1. Application Overview

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

* The web based ERM application that users interact with.
* An ArcGIS Enterprise deployment that hosts ERM data and controls access to application.
* A dedicated server that hosts the ERM API middleware.
* A service that allows ERM to integrate with the business system of record.
  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).
* Node machine to hold ERM middleware API.
* Web Server to host ERM Application

The *ERM Environment Setup Guide* has instructions for setting up entire environment.

See Appendix A in this doc for a readiness checklist.

* 1. File Prerequisites

The following files should have been provided from Esri for deployment of the application:

* Folder with Pro project and files for publishing ERM Feature Services
* Folder with files for installing ERM Middleware API.
* Folder with the built ERM Web Application
* Optional:
  + Folder with files to deploy a sample Business System Integration service

1. ArcGIS Enterprise
   1. Portal Items

ERM requires a few items in Portal: Web Map, Web App, Groups for Locations.

These items can either be shared with your entire organization, or you can create a group just for ERM items and users.

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.

* + 1. Web Map for App

Create a new web map for the application to use.

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. Basemap does not matter, application configuration will set that.
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)
   * 1. Application Item

Create a web application item in Portal for routeplanner.

1. Follow the instructions [here](https://developers.arcgis.com/web-appbuilder/guide/getstarted.htm) under "Provide an app ID for Web AppBuilder" steps 2-6, using the URL where the app will be deployed.
2. Give name RoutePlanner.
3. Share the application item with your Organization or a defined ERM group.
4. Save the item ID found in the URL.
5. Save the registered ID for the item. Found on the Settings page under Registered Info. Will be shorter than the item ID.
   * 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 All Items, not just their own.
6. Add users to group.

A ERM\_CreateGroups.py script is available in the [GitHub document repository](https://github.com/EsriPS/enterprise-route-management/tree/master/Operational-Documentation). 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 must be member to Group for any Location that they plan.
* User account has either a Publisher or Administrator Role set.
* If using a group to share items rather than the organization, User must be member of group.

1. ERM Feature Services

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

The same user that was used to create Portal items in section 3.0 should be used to publish the feature services.

* 1. Sample Data

In the Pro project there are maps with “\_Sample” suffix. These maps point to geodatabases in the fgdbs\_Sample folder. The tables and feature classes contain sample location data, to give examples of how to populate the data when loading in your location information.

If these sample services are published it is recommended to leave the “\_Sample” suffix on the name and update the names in the middleware and web app configurations.

This data only needs to be published if using the sample Business System Integration Service.

* 1. Load Customer Data

This step can be done on any computer with ArcGIS Pro installed and access to the ERM Portal and Server. Loading any new data should be done to the geodatabases in the fgdbs folder and use the maps without a “\_Sample” suffix.

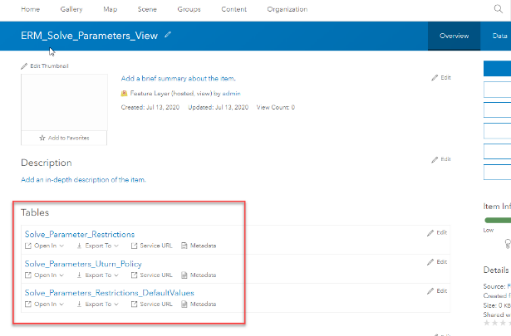
**Note**: It is recommended that the layers be left in the same order that they are presented. The application expects the index value of some layers to be certain value. These can be updated in the configuration files but leaving in default order will save from having to update those values.

1. In the extracted delivery directory, open the services folder.
2. Open the ERM\_services Pro project included with release.
3. Verify the following maps are available and have no broken links.
   1. ERM\_Plan\_Defaults
   2. ERM\_Plan\_Template
   3. ERM\_Registry
   4. ERM\_Solve\_Parameters
   5. MDM\_Locations
   6. GPS\_Template
4. Open the ERM\_Plan\_Defaults map. Verify these layers and update with data for your locations.
   1. DepotTemplate - add a point feature for each location.
      1. Core Pro tools can be used to import locations from other databases. See online help documentation [here](https://pro.arcgis.com/en/pro-app/help/data/geodatabases/overview/import-data.htm).
   2. LineBarrierLibrary – add linear features that act as a barrier (optional).
   3. PolygonBarrierLibrary – add polygon features that act as a barrier (optional).
   4. ZoneTemplate – add polygon features that define zones (optional).
   5. BreakTemplate table – add break information for each route (optional).
   6. DispatchLocation table – add information for each location.
   7. RouteTemplate table – add all available routes for all locations.
   8. SpecialtyNameTemplate table – enter information on available specialties (optional).
5. Use the ERM\_Plan\_Template map for display settings of the layers.
   1. For all the layers except Routes, set the symbology for how they will display in the ERM application.
   2. The default symbology for Routes is generated when Plan is created. The symbol shown in map is not used.
   3. Field order and visibility will be set in the web application configuration file.
6. ERM\_Registry map
   1. No update needed. This is used by the application to store a list of all plans.
7. 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.
8. MDM\_Locations
   1. Add point feature for each location.
      1. Core Pro tools can be used to import locations from other databases. Or Append from the Depot Template layer in ERM\_Plan\_Defaults. See online help documentation [here](https://pro.arcgis.com/en/pro-app/help/data/geodatabases/overview/import-data.htm).
   2. These locations are used to populate a pick list to select locations to move orders to.
9. GPS\_Template
   1. No update needed. This is a container used to show current location of trucks if configuring the system for that functionality.
10. Make sure all edits are saved and save the Pro project.
    1. Batch Publish

If publishing all maps at once, scripts are available to batch the process. If publishing the Sample data, check the “Use Sample” checkbox on the tools.

1. Open the ERM\_services Pro project included with release that was used to populate the data.
2. From Catalog View, open the ERM\_services toolbox.
3. Open the Create ERM SD files tool.
4. Check “Create All” and run tool.
   1. This will create service definition files in the sd\_drafts folder.
   2. If Use Sample is checked, the sample data will be used to create sd files.
5. Open the Publish SD Files tool.
   1. Enter the Portal information and run tool. Use a publisher or admin level user.
   2. Check the Use Sample box if publishing sample data.
6. Verify in Portal that an ERM Services folder is created and contains the following items (there will also be a service definition file for each layer):
   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
      1. See note below on validating view.
   6. MDM\_Locations feature layer
   7. GPS\_Template feature layer

**Note**: 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 section 4.4.



* 1. Manual Publish

If you want to replace all the services, can just delete existing and re-run the procedure for Initial Deployment. If a single service needs to be republished, these steps can be followed.

1. Open the ArcGIS Pro project
2. In Pro, sign into the Portal that the services will be published to.
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

**NOTE**: If the ERM\_Registry service is deleted, existing plans will no longer be available in the ERM Route Planner application.

* + 1. Updating - Overwrite Existing

If the data services need to be updated, the existing services can be overwritten. This could be for adding locations or making symbology changes. This is only valid for the feature layers and not table layers: ERM\_Plan\_Defaults, ERM\_Plan\_Template, and MDM\_Locations. For Table layers need to delete the existing service and republish.

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.
11. ERM Middleware
    1. Initial Install

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. 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.
5. Open a command prompt with run as Administrator option and navigate to the middleware directory.
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*
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. Updating the Middleware

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.

Follow these steps when having to redeploy the API service due to an update.

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 initially 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
5. Run install steps from section 5.1 using new scripts and middleware folders
6. Web Application
   1. Initial Install

The ERM web application needs to be deployed to web server. In the instructions below the folder is called “routeplanner”, but it can be changed if a different URL will be used.

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://[servername]/routeplanner and the web application should load.
   1. Updating the Web Application

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.

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 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.

NOTE: 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. 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.
10. Run the Publish\_BSI\_TestService.py script.
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 tool 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
13. From the extracted python folder, copy the config folder into the p20 folder.
14. Can open Server Manager and verify item is published and is started.
15. Stop and Restart the BSI service to verify it picks up the configuration update.
16. Sanity Test

After all deployment steps are complete, run the following procedure as a sanity test.

1. Log into Route Planner app.
2. Click Create New Plan.
   1. 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 1/1/2020 12:00.
3. Verify correct list of locations are listed.
4. Choose a new location, set date and time for the data available, and create new plan.
5. 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.
6. Open Routes tab.
7. Click Run Route Optimization.
8. Routes are solved and geometries assigned.
9. 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 | Publish feature services with sample data |  |
| 2 | Publish sample BSI Service |  |