

Config Exporter

- 1. Use case
 - 1.1 Download
 - 1.1 API Support
 - 1.2 Controller Compatibility
- 2. Server Startup
 - 2.1 Server Port
 - 2.2 Authentication
 - 2.3 Proxy Support
 - 2.4 SSL Support
- 3. Usage
 - 3.1 Add Controllers
 - 3.1 Export Configuration
 - 3.2 File Download
 - 3.3 File Upload
- 4 Supported Configurations
- 5 Entity Source
- 7 Encryption
- 8 Support

1. Use case

Config exporter can be used to migrate the configuration between controllers or applications. The configuration can be imported directly into another controller/application or it can be download as a file.



1.1 Download

The tool is available for download from <https://tools.appdynamics.com>.

1.1 API Support

Please refer to [Config Exporter API](#) for details on the supported API.

1.2 Controller Compatibility

The config exporter is not cross compatible with all versions of controller. Please refer to the following table and choose the appropriate version.

Controller Version	Config Exporter Version	Download
4.0x	1.x	Download
4.1x	2.x	Download
4.2x	4.2x	Download
4.3x	4.3x	Download

4.4x	4.3x	Download
4.5.0	4.5.2	Download
4.5.1+	4.5.2.1+ 4.5.7.1+	Download

Note: If there are any major non backward compatible changes in the new release controller apis / model , then config exporter will be incompatible with that version. The fix would require a code change and release of the config exporter.

2. Server Startup

The tool is shipped as a lightweight java application with an embedded server. To start the application, all you need is to type the following command, after replacing `$version` with the latest version. The following command will start Config Exporter on port 8080 by default

```
java -jar config-exporter-$version.war
```

Java Version: Java 1.8+ is required from Config Exporter version 4.3.17 onwards

2.1 Server Port

The default port is 8080. To use a different port like 8181

```
java -Dserver.port=8181 -jar config-exporter-$version.war
```

2.2 Authentication

Config exporter is mostly used as a single user app, so it stores the encrypted credentials in the file system for that user. To make it available for multiple users, please enable authentication. If the authentication is enabled, config exporter will prompt for credentials when a controller is selected from the dropdown. The credentials will not be stored in this case.

```
java -Duser.authentication.enabled=true -jar
config-exporter-$version.war
```

2.3 Proxy Support

Please make sure that you add both the `http` and `https` properties. Username and password are optional, it can be omitted for a non-authenticating proxy

```
java -Dhttp.proxyHost=<host> -Dhttp.proxyPort=<port> \
-Dhttps.proxyHost=<host> -Dhttps.proxyPort=<port> \
-Dhttp.proxyUser=<user> -Dhttp.proxyPassword=<password> \
-jar config-exporter-$version.war
```

2.4 SSL Support

This will start the server on ssl port 8443.

```
java -Duser.authentication.enabled=true \
-Dserver.port=8443 \
-Dserver.ssl.key-alias=key-alias-name \
-Dserver.ssl.key-store=file:/full/path/to/keystore.jks \
-Dserver.ssl.key-store-password=secret \
-jar config-exporter-$version.war
```

3. Usage

Please open a browser and navigate to <http://localhost:8080> or wherever you have started the server from. Please note that only HTML5 enabled latest browsers are supported.

3.1 Add Controllers

The next step is to add the controllers. Click on the + icon next to the controller to add a controller. Once it is saved, the controller will appear in the dropdown.

3.1 Export Configuration

This step is same for all the configurations. The webpage is divided into 3 sections

1. **Left Pane:** This is the source details where to export from and what entities are supported
2. **Center Pane:** This lists the entities and user can selectively import the configurations into the destination controller
3. **Right Pane:** This is the destination details, like the destination controller, destination application etc.

3.1.1 Source

1. Select a source Controller
2. Select an Application
3. Select the Tier (Only required for Service Endpoints)
4. Click on one of the configurations

3.1.2 Entity Selection

In the Center pane, select the **Entity Source** (Application or Tier) by clicking on it. Only the Tiers where the configuration is overridden is shown here. Application is selected by default. Click on the Tier Name to select the tier. Please refer the **Entity Source** section for more details.

APPLICATIONS > App1 > ENTRY POINTS



Then select the entities that you want to import into the destination and click "Next" button at the bottom. This will show the destination section

3.1.3 Destination

1. Select the destination Controller. This can be same as source controller, if you are importing configuration from one Application to another
2. Select a Destination Application
3. **Destination Tier:** Some configuration can be imported to the Tier level also. For example, Custom Entry Points, Backend configuration, Service Endpoints etc.
4. Click on the ">" button next to the Application to bring up the export window.

3.1.4 Export Window

Here you can review the configurations, and click on the "Export". It will push the configurations and show the status. Some of the status values are [hyperlinked](#), please hover the mouse over to see the details. The following are some of the status values

1. ALREADY EXISTS: You can choose to overwrite the existing entities by selecting the **overwrite** checkbox.
2. NOT_FOUND: It means some dependent entities are not found. Hover over to get the details.
3. NEED_DETAILS: Hover over to get the details.

3.2 File Download

The configurations can be downloaded as file and can be later uploaded to the tool. To download the file, click the download button in the center pane. This will be at the bottom section below the list of entities.

3.3 File Upload

The same file can be uploaded and imported into the controller. Select the "File Upload" tab in the Left Pane and drop the file into the section. You can also choose the button to upload the file. Once the file is uploaded then the, import process is same.

4 Supported Configurations

As of version 4.5.2-BETA, the following are supported

1. TRANSACTION DETECTION
2. CUSTOM ENTRY POINTS (CUSTOM MATCH RULES)
3. CONFIG 2.0 SCOPES
4. CONFIG 2.0 RULES
5. BACKEND DETECTION
6. CUSTOM EXIT POINTS
7. INFORMATION POINTS
8. SERVICE END POINTS
9. SERVICE ENDPOINT DETECTION
10. BT CONFIG
11. HEALTH RULES
12. ACTIONS
13. POLICIES
14. DATA COLLECTORS
15. CALL GRAPH SETTINGS
16. ERROR DETECTION
17. EUM CONFIGS
18. JMX RULES
19. APPAGENT PROPERTIES
20. SLOW TRANSACTION THRESHOLDS
21. METRIC BASELINES
22. EUM APP INTEGRATION
23. DB COLLECTORS
24. SERVICE AVAILABILITY MONITORING
25. ADMIN SETTINGS
 - a. ROLES
 - b. GROUPS
 - c. USERS
 - d. AUTHENTICATION PROVIDER
26. EMAIL TEMPLATES
27. HTTP TEMPLATES
28. EMAIL SMS CONFIG
29. DASHBOARDS (via ACCOUNT EXPORT)
30. ASYNC TRANSACTION CONFIGURATION (v4.3.23)
31. LICENSE RULES (v4.3.27)

The version 2.x+ also supports the export / import of Application configurations. This way all the configurations can be exported in one shot. This is not currently available for 1.x version. The 4.3.18+ supports the export / import of account configurations.

5 Entity Source

Some of the entities like Transaction Detection, Custom Entry Points etc. supports hierarchical configuration, which means configuration can be created at Application level or at Tier level. Config exporter supports the export between any of these entities i.e App App | App Tier | Tier Tier | Tier App. Please refer to the screenshots while selecting the right configuration. In the export popup, always review the

SOURCE and DESTINATION sections prior to export.

Scenario 1: Application to Application Export

APPLICATIONS > App1 > ENTRY POINTS

App1

App1_Tier1

App1 > ENTRY POINTS

☐ Name

☒ Cron4J

☐ JavaTimer

☐ JCronTab

☐ Quartz

Type

POJO

POJO

POJO

POJO

Enabled

✗

✗

✗

✗

Next >

DESTINATION CONTROLLER

CONTROLLERS +

http://192.168.1.132:8090

ENTRY POINTS DESTINATION

☒ APPLICATION

☐ TIER

The ENTRY POINTS configurations will be exported to the selected Applications

APPLICATIONS +

App2

Use regex to match ☐

APPLICATION -> APPLICATION

IMPORT CUSTOM ENTRY POINTS

SOURCE

CONTROLLER

http://192.168.1.132:8090

APPLICATION

App1

DESTINATION

CONTROLLER

http://192.168.1.132:8090

APPLICATIONS

App2

Export

CUSTOM ENTRY POINTS (1)

NAME	TYPE	DEST APP	<input type="checkbox"/> OVERWRITE	RESPONSE STATUS
Cron4J	POJO	App2	<input type="checkbox"/>	

Close

Scenario 2 : Tier to Tier Export

APPLICATIONS > App1 > ENTRY POINTS

App1

App1_Tier1 ✓ 1

App1 > TIERS > App1_Tier1 > ENTRY POINTS

✓ Name	Type	Enabled
✓ Cron4J	POJO	✗
✓ JavaTimer	POJO	✗
✓ JCronTab	POJO	✗
✓ Quartz	POJO	✗

Next > Download

TIER > TIER

DESTINATION CONTROLLER

CONTROLLERS +

http://192.168.1.132:8090

ENTRY POINTS DESTINATION

APPLICATION TIER 2

The ENTRY POINTS configurations will be selected Tiers. Select the Application and the select the tiers to export it to.

APPLICATIONS +

Use regex to match

✗ App2

TIERS

Use regex to match

✗ Tier1 3

IMPORT CUSTOM ENTRY POINTS

SOURCE

CONTROLLER http://192.168.1.132:8090

APPLICATION App1

TIER App1_Tier1 4

DESTINATION

CONTROLLER http://192.168.1.132:8090

TIERS App2 > Tier1 5

Export

7 Encryption

As of version 2.0.6+ and 1.6.8+, the controller credentials encrypted and saved as a dat file. The file is located in a directory named .cache as the same directory as the war file. It uses a default encryption key. To use a different encryption key use the system property -Dencryption.key=*** at the startup. The application doesn't currently support the key migration, if you want to change the key then delete the .cache directory and restart the application with the new key. You will have to add the controller info again.

8 Support

Please contact help@appdynamics.com for any questions. The support will be on a best effort basis.