



# SSF Tools: Object Exporter User Guide

---

Version 3.1

## Document Revision History

Revision Date	Written/Edited By	Comments
May 2015	Paul Wheeler	Initial Release
June 2016	Paul Wheeler	Minor additions and changes for release 1.1
August 2016	Paul Wheeler	Details for CDATA and Merge file options in release 2
June 2017	Paul Wheeler	Minor update for version 2.1, SSD v4
September 2017	Paul Wheeler	New option for pattern-matching object names for export, for version 3.0, SSD v5.
May 2018	Paul Wheeler	Version 3.1, SSD v6: added option to strip metadata that is typically not useful when migrating objects between environments.

© Copyright 2017 SailPoint Technologies, Inc., All Rights Reserved.

SailPoint Technologies, Inc. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

**Restricted Rights Legend.** All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

**Regulatory/Export Compliance.** The export and reexport of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or reexport outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Government's Entities List; a party prohibited from participation in export or reexport transactions by a U.S. Government General Order; a party listed by the U.S. Government's Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

**Trademark Notices.** Copyright © 2017 SailPoint Technologies, Inc. All rights reserved. SailPoint, the SailPoint logo, SailPoint IdentityIQ, and SailPoint Identity Analyzer are trademarks of SailPoint Technologies, Inc. and may not be used without the prior express written permission of SailPoint Technologies, Inc. All other trademarks shown herein are owned by the respective companies or persons indicated.

## Table of Contents

Overview .....	4
Installation.....	5
Using the Object Exporter .....	5

## Overview

The Object Exporter is a custom IdentityIQ task for quick and easy export of XML objects such as applications, rules, workflows, configurations and any other objects that are managed in IIQ. The task has the following features:

- Exports XML objects to the same filesystem structure used by the Services Standard Build (SSB)
- Allows the user to specify which classes get exported
- Allows the user to specify which objects get exported by matching their names with a regular expression
- Can optionally export only the objects that have been created or modified after a given date
- Can optionally remove IDs and timestamps from the exported objects
- Can optionally remove metadata that is typically not useful when migrating objects between environments
- Allows definition of a naming format for the exported XML files
- Can optionally produce tokenized XML files ready for use in the SSB build, using reverse-lookup to the tokens in a target.properties file
- Can add CDATA sections and unescape BeanShell code and other escaped elements
- Can create “merge” files by comparing the XML of objects being exported with previously exported base versions of the objects (for supported object types).

The tool is designed to help developers working in an IdentityIQ sandbox or development environment, particularly when working directly within IIQ rather than working on external master files and importing those. When working directly within IIQ it can be easy to lose track of what was changed and when, making it difficult to keep the build up-to-date. The Object Exporter provides the following advantages over other export methods:

- Allows the developer to easily export everything they have been working on over a given period, or since the start of a project, removing the need to maintain scripts used to export objects from the console.
- By maintaining a standard naming format for the resulting files it also helps to prevent the creation of multiple copies of the same object XML when there is more than one developer working on an IdentityIQ project.
- Avoids the need to tediously replace text in the XML files with the tokens used in the SSB.
- Avoids the need to create the SSB folder structure when exporting objects.
- The resulting XML files can be added directly to the SSB since they are in the correct folder structure and have the correct tokens.
- The simple process also makes it easy to provide exported files that can be uploaded to a code repository or just stored outside of IIQ as a quick and simple backup of everything that has changed.
- Fixes issues that the console checkout process has with exporting IDs rather than names for objects referenced by certain classes such as IdentityTrigger and Scope.
- The use of CDATA sections with unescaped code aids the readability of the code when viewed offline.
- Allows the export of “merge” files that contain only the additions and changes in certain types of objects when compared with base exports.

## Installation

The Object Exporter consists of a Java class (ExportXML.class) and a Task Definition XML file (TaskDefinition\_Export\_XML.xml). To install, follow these steps:

- Shutdown the application server
- Add ExportXML.class to WEB-INF/classes/sailpoint/services/task/
- Restart the application server
- Import the Task Definition xml file in the console using the “import” command, or alternatively in the UI using the “Import from File” functionality under Global Settings (IdentityIQ 7.0 and later) or System Setup (previous versions).

If you are creating a build using the SSD (Services Standard Deployment), the class file will automatically be added to the correct location and the Task Definition will be imported as part of the normal import of custom objects.

## Using the Object Exporter

To access the Object Exporter in IdentityIQ, go to Monitor -> Tasks. Create a new task of type “XML Object Exporter”. The task has several user-configurable options:

### Standard Properties

\*Indicates a required field.

**Name\***

**Description**

**Allow Concurrency** ☐

**Require Signoff** ☐

**Host**

**Number of Runs**

**Average Run Time**

[Reset Run Statistics](#)

### Email Task Alerts

**Email Notification**

Previous Result Action

### XML Object Exporter Options

Base path for export*	<input type="text" value="d:\iiqexport\config"/>
Remove IDs from exported XML	<input checked="" type="checkbox"/>
Add CDATA sections	<input checked="" type="checkbox"/>
Classes to export (leave blank for all, use 'default' for a default set of classes)	<input type="text" value="default"/>
Only include objects with names matching this regular expression	<input type="text" value="MyCorp*"/>
Strip environment-specific metadata	<input checked="" type="checkbox"/>
Only include objects updated or created after this date	<input checked="" type="checkbox"/> 4 June 2018 08:00 AM
Naming format (see help text for variables)	<input type="text" value="MyCorp_\${NAME\$}"/>
target.properties file for reverse-tokenization for SSB	<input type="text" value="d:\mybuild\dev.target.properties"/>
Directory containing original XML files for comparison when creating merge files	<input type="text" value="d:\BaseFilesForMerge"/>

**Base path for export**

Enter the path to which you want to export the XML files. This must be a path that can be accessed by the server that runs the task.

**Remove IDs from exported XML**

Defines whether the exported XML files will include the IDs and created/modified timestamps for the objects in the environment that you are exporting from. In most cases this should be set to True.

**Add CDATA sections**

Encloses code in CDATA sections and unescapes the code for ease of reading in the exported file. This works for BeanShell code in rules and scripts, HTML formatting in email templates and SOAP messages in IntegrationConfig objects

**Classes to export**

This should be a comma-separated lists of object types to export. If left blank it will export all object types. If set to “default” it will export only the types of objects that are commonly exported during an IdentityIQ project. It can also be a combination of “default” and other classes – for example you could use “default,Scope”.

**Only include objects with names matching this regular expression**

Specify a regular expression if you want to only export objects with names matching a given pattern. For example, if you just want to export all objects that start with “MyCorp” you would enter “MyCorp.\*”. Leave this blank if you do not wish to filter by object name.

**Strip environment-specific metadata**

Remove data from objects that is not typically useful when migrating objects between environments. The following data will be removed:

- Statistical data on TaskDefinitions, such as number of runs and average run time
- Timestamps on TaskSchedule objects for last and next execution
- Entries in Application objects for account aggregation start and end timestamps, delta aggregation cookie data for Active Directory applications

**Only include objects updated or created after this date**

If enabled, this option allows the user to enter a date and time which defines the earliest creation and modification timestamps by which we will filter the objects to be included in the export. If disabled, creation and modification timestamps will be ignored.

**Naming format**

Allows the user to define a format for naming the exported XML files. There are two optional variables that can be used:

- `$Name$`        The name of the XML object
- `$Class$`        The name of the object class, e.g. Application, Rule etc.

Using these variables, you can build a custom naming format for the exported files. For example, “MyCorp\_ \$Class\$ \_ \$Name\$” would export an application called “ActiveDirectory” with the file name “MyCorp\_Application\_ActiveDirectory.xml”. If this option is left blank the XML file will be named as the object name with a “.xml” extension.

Illegal characters (anything that is not an alphanumeric character, dot or hyphen) and spaces will be replaced with underscores in the file names.

**target.properties file for reverse-tokenization for SSB**

Optionally defines the path to a target.properties file that provides token values for the sandbox/development environment in the usual SSB format. The task performs a “reverse-lookup” on any text it finds in the XML that matches a token value found in the target.properties file, and replaces it with the corresponding token. If this option is left blank, no tokenization of the export files will take place. Note that tokens in the target.properties file that have values of “true” or “false” will be ignored in the current version of the Object Exporter since these values are present in many objects and replacing them all with a token is not usually the intention. The same applies to token values that are blank or just whitespace.

**Directory containing original XML files for comparison when creating merge files**

Optionally defines the location under which the “base” XML files can be found for use when creating “merge” files. These are files that contain only the XML entries that have been added or changed when compared with the base files and will be exported with the ImportAction property set to “merge”. The task compares the current XML being exported with the base files and creates merge files based on the differences. When imported into IdentityIQ, the entries in these files will be merged with an existing object that has the same name, rather than overwriting it. Merge files should be used for certain types of objects (such as Configuration and UIConfig objects) where an upgrade to IdentityIQ may result in the addition of new entries in these objects, which would otherwise get removed when importing a full object that does not have an ImportAction of “merge”.

At the start of a project it is useful to export all the objects that support merging to obtain the base files that are referenced here and place these in a single location. When looking for merge files in the provided location, the task will recursively search in subfolders for any files that have .xml extensions and use these as the base files.

The task currently supports objects of the following types for merging:

- Configuration
- UIConfig
- ObjectConfig
- AuditConfig
- Dictionary

After selecting the desired options, click “Save and Execute”. The task will start running and its progress can be monitored in the Task Results. When completed, the results will show a summary of the number of XML files exported for each class and a total of all objects exported.