infotel

Arcsys BatchReporting

User Guide

25.3.STS
October 30, 2025

infotel.com

Version: 25.3.STS

Publication date of the document: 2025-10-30

This document is intended for anyone who requires more information on the Arcsys product.

This document is written for all owners of a valid license.

The reader agrees to respect the confidential nature of this document.

This document is part of the Arcsys software package, a product designed and developed by Infotel. All rights are reserved for Infotel.

Contact details:

France	***Germany***	***USA***
INFOTEL SA *Le Valmy, 6/8/18 Avenue Léon Gaumont* *F-75020 Paris* *France*	*Insoft Infotel Software GmbH* *Sternstr. 9-11* *D-40479 Düsseldorf* *Deutschland*	*INFOTEL Corporation* *PO Box 47517 Florida* *33743 St Petersburg* *United States*
+33 (0)1 48 97 38 38	*+49 (0) 211 44 03 16-6*	*800 543 1982 – Toll-free telephone (US only)* *+1 727 343 5958*
https://techsupport.infotel.com *software@infotel.com*	*https://techsupport.insoft-software.com* *software@insoft-infotel.com*	*https://techsupport.infotel-consulting.co.uk* *software@infotel.com*

Copy prohibited without explicit authorization. © 2025 Infotel SA All rights reserved.

Table of Contents

Preface 5
1. Introduction 5
2. Reference Documents 5
2.1. Installing and Updating 5
3. Symbols and Meanings 5
4. Definitions and Abbreviations 5
1. Getting started with BatchReporting 1
2. Installing BatchReporting 3
1. Prerequisites 3
2. Installing the module 3
3. Finalizing installation 3
3.1. Configuring BatchReporting 3
3.2. How to encrypt passwords in BatchReporting configuration files 3
3. Updating BatchReporting 6
4. Using BatchReporting 7
1. Launching BatchReporting 7
2. Filtering the report 7
2.1. The type of report 7
2.2. The type of entity 7
2.3. The types of request 11
2.4. The time period 11
3. Customizing the report 11
4. Automatic resumption of an interrupted report 12
4.1. Algorithm description 13
4.2. Deactivating the automatic resumption of the report 13
4.3. Resume files location 13
5. Module description 15
1. Directory structure 15
2. etc/<global_configuration_file>.properties file 15
3. etc/http-client.properties file 19
3.1. HTTP connection settings 19
4. etc/log4j2.xml file 22
Glossary 23
Registered Trademarks 30

List of Tables

4.1. Examples of configuration to filter on repositories 8
4.2. Examples of configuration to filter on collections 8
4.3. Examples of configuration to filter on classes 9
5.1. Content of the `batch-reporting` directory 15

Preface

1. Introduction

This document provides detailed information on how to use Arcsys BatchReporting.

2. Reference Documents

2.1. Installing and Updating

Arcsys Prerequisites Manual: **Arcsys-requirements-25.3.STS-en.pdf**

Arcsys Installation Manual: **Arcsys-installation-25.3.STS-en.pdf**

3. Symbols and Meanings

Note

Identifies information of particular interest

Important

Identifies important information

4. Definitions and Abbreviations

See the page 23, « Glossary ».

1. Getting started with BatchReporting

BatchReporting allows you to generate an Arcsys usage report.

This report contains Arcsys usage statistics in number of requests (ended successfully or in error) associated to cumulative sizes.

BatchReporting allows you to filter the reports on the type of the report, the type of request, the type of entity and a specific time range. You can customize the report so that it concerns only one or several entities of the same type (repository, collection or class), one or several request types (archiving, synchronous retrieval, retrieval requests...) and a particular period (see page 7, « Filtering the report » for more details).

The generated report consists of a CSV document. This document presents the number of requests:

- monthly or daily
- ended successfully or in error
- for a defined period
- for specific repositories, collections or classes
- for each specified request types

The CSV may also include the cumulative size of requests ended successfully, either on a monthly or daily basis.

In the following example, the CSV presents the number of monthly requests, ended in error, of `archive` type, for the `base2` repository. It is possible to specify multiple types of requests as well as multiple repositories, collections or classes.

```
App property file name=batch\src\test\resources\etc\example.properties
httpclient.properties.file=batch\src\test\resources\etc\http-client.properties
report.type=count requests errors
entity.type=repository
1.repository=base2
request.types=archive
start.date=2015/01/01 00:00:00
end.date=2030/12/31 23:59:59
group.by=MONTH
MONTH;base2 \ archive in error
2022/04;1
2022/06;2
2022/12;1
2023/01;15
2023/03;15
2023/04;1
2023/11;2
```

Opening the file in a spreadsheet will yield the following result:

	A	B
1	App property file name=batch\src\test\resources\etc\example.properties	
2	httpclient.properties.file=batch\src\test\resources\etc\http-client.properties	
3	report.type=count requests errors	
4	entity.type=repository	
5	1.repository=base2	
6	request.types=archive	
7	start.date=2015/01/01 00:00:00	
8	end.date=2030/12/31 23:59:59	
9	group.by=MONTH	
10	MONTH	base2 \ archive in error
11	2022/04	1
12	2022/06	2
13	2022/12	1
14	2023/01	15
15	2023/03	15
16	2023/04	1
17	2023/11	2

You can customize the content of this output file, as well as its formatting (see page 11, « Customizing the report » for more details).

2. Installing BatchReporting

1. Prerequisites

The prerequisites for the module align with the prerequisites of Arcsys as detailed in Arcsys Prerequisites Manual.

2. Installing the module

To install the module, please refer to the Arcsys Installation Manual.

3. Finalizing installation

3.1. Configuring BatchReporting

After extracting BatchReporting, the configuration files must be set with the values that depend on your environment.

Some parameters are used to contact the Arcsys REST API. To set those parameters, please refer to page 19, « etc/http-client.properties file » chapter.

Others parameters are used to specify the content of the report. To set those parameters, please refer to the page 7, « Filtering the report » and page 11, « Customizing the report » chapters.

3.2. How to encrypt passwords in BatchReporting configuration files

Note

The encrypt module is installed in the root folder of Arcsys Engine or present in the Arcsys installer.

The module can be used to encrypt any value of BatchReporting (and not only passwords). When a parameter value beginning with `ENC(` is encountered, BatchReporting automatically decrypts it.

3.2.1. Generate key file

In order to encrypt the data, it is necessary to generate first a key file with the `generate-key` script present in the Arcsys installer. This script uses a cryptographically secured pseudo-random number generator to create a random 256 bit symmetric key.

The key file is used by Arcsys to decrypt the encrypted passwords in the configuration files of different modules. This key file may be shared across modules (as it must be different each time, as required by your security policy).

The following GNU/Linux command generates a file (filename: `secret-key`) in the folder `myFolder`:

```
$ ./generate-key.sh -s -d myFolder
Secret key generated in file myFolder/secret-key
```

3.2.2. Encrypt a value

The passwords must be encrypted with the generated symmetric key file. All properties can be encrypted too.

The command for using the *encrypt* utility depends on the platform chosen:

- Windows: `bin\encrypt.cmd`
- GNU/Linux: `bin/encrypt.sh`

The following GNU/Linux command encrypts the password `myPassword` with the `secret-key` file:

```
$ ./encrypt.sh -k myFolder/secret-key -e
Please enter the value to be encrypted.

Please enter the value again.

ENC(fRO8S6BlhZY5Ci3Rn8ZJgOXWkwD5jYgaYa3QWBsV52Q=)
```

The blank line corresponds to the entry of the string to encrypt.

Note

If the option `-k myFolder/secret-key` is not specified, the default key stored in `etc/secret-key` is used.

The encrypted values (including "`ENC(`" and "`)`") of the password must then be copied in the configuration file.

The encrypted properties are *salted*, in the cryptographic sense, and associated with the key present in the `secret-key` file. If this key is renewed, all the properties encrypted using a previous key must be re-encrypted manually.

3.2.3. After installation

The key file must be saved in the `etc/` directory of BatchReporting installation directory. Its name must be `secret-key`.

The secret key file may be installed in another directory and/or under another name. In this case, you have to manually update the `bin/overenv.sh` (or .cmd) script. The `SECRET_KEY_FILE` parameter must be set to specify the path and name of the secret key file. For example:

```
SECRET_KEY_FILE=/opt/Infotel/batch-reporting/secret-key
```

3. Updating BatchReporting

To update the module, please refer to the Arcsys Installation Manual.

4. Using BatchReporting

1. Launching BatchReporting

The script is used as follows:

```
./batch-reporting.sh -p <configuration_file>
```

or, on Windows platform:

```
batch-reporting.cmd -p <configuration_file>
```

The configuration file corresponds to the file where all the main parameters of BatchReporting are set.

It is recommended to save this file in the `etc/` directory of BatchReporting installation directory.

See page 15, «etc/<global_configuration_file>.properties file » for more details.

2. Filtering the report

This part presents how to select the information to be included in the report:

- the type of report
- the type of entity
- the types of request
- the time period

These selection criteria are provided in the form of parameters, which are specified in the main configuration file of BatchReporting.

Note

Only the entities that the user can see will be considered in the report.

2.1. The type of report

There are two possible types of report: “`count-size requests ended` ” and “ `count requests errors`”. The type of report is selected with the *`report.type`* parameter.

2.2. The type of entity

The type of entity concerned by the report is selected with the *`entity.type`* parameter. The type of entity that can be selected is limited by the *`report.type`* chosen.

- When *report.type*=`count-size requests ended`, you can only select `repository`, `collection` or `class` entities.

- When *report.type*=`count requests errors`, you can only select `repository` type.

Once the entity type chosen, you have to specify the codes of the entities chosen for the report. For each entity type, you can find below some examples of configuration.

2.2.1. Filtering on repositories

To filter the report for specific repositories, set the *entity.type* parameter to `repository` . Then, specify the repositories that are concerned by the report:

Number of repositories concerned by the report	Associated configuration
One repository	`1.repository=<repository code>`

n repositories	`1.repository=<repository code>`
	`2.repository=<repository code 2>`
	`...`
	`n.repository=<repository code n>`
	or
	`1.repository=<repository code>;<repository code 2>;...;<repository code n>`

All allowed repositories	`1.repository=*`

Table 4.1. Examples of configuration to filter on repositories

2.2.2. Filtering on collections

To filter the report for specific collections, set the *entity.type* parameter to `collection` . Then, specify the collections that are concerned by the report by using the pair <repository code>/<collection code>.

Each selection is composed of two lines. Those lines are prefixed by a number, identifying the selection. The first line defines the repository, and the second line defines the collection.

Here are some examples of configuration:

Number of collections concerned by the report	Associated configuration
One collection	`1.repository=<repository code>` `1.collection=<collection code>`

Number of collections concerned by the report	Associated configuration

n collections	`1.repository=<repository code 1>`
	`1.collection=<collection code 1>`
	
	`2.repository=<repository code 2>`
	`2.collection=<collection code 2>`
	
	`…`
	
	`n.repository=<repository code n>`
	`n.collection=<collection code n>`

n collections of the same repository	When many collections are selected in the same repository, a list of collections may be used to select them:
	
	`1.repository=<repository code>`
	`1.collection=<collection code 1>;<collection code 2>;... ;<collection code n>`

All allowed collections of a repository	`1.repository=<repository code>` `1.collection=*`
All allowed collections of all allowed repositories	`1.repository=*` `1.collection=*`

Multi criteria	`1.repository=<repository code 1>`
	`1.collection=<collection code 1>`
	
	`2.repository=<repository code 2>`
	`2.collection=<collection code 2>;<collection code 3>;…;<collection code n>`
	
	`3.repository=<repository code 3>`
	`3.collection=*`
	
	`…`

Table 4.2. Examples of configuration to filter on collections

2.2.3. Filtering on classes

To filter the report for specific classes, set the *`entity.type`* parameter to `class`. Then, specify the classes that are concerned by the report by using the pair <repository code>/<class full path>.

Each selection is composed of two lines. Those lines are prefixed by a number, identifying the selection. The first line defines the repository, and the second line defines the full path for the class.

Here are some examples of configuration:

Number of classes concerned by the report	Associated configuration
One class	`1.repository=<repository code>` `1.class=<class full path>`

Number of classes concerned by the report	Associated configuration

n classes	`1.repository=<repository code 1>`
	`1.class=<class full path 1>`
	
	`2.repository=<repository code 2>`
	`2.class=<class full path 2>`
	
	`...`
	
	`n.repository=<repository code n>`
	`n.class=<class full path n>`

n classes of the same repository	`1.repository=<repository code 1>`
	`1.class=<class full path 1>;<class full path 2>;...`
	`;<class full path n>`

All classes in a repository	`1.repository=<repository code>` `1.class=*`
All classes in all repositories	`1.repository=*` `1.class=*`

Multi criteria	`1.repository=<repository code 1>`
	`1.class=<class full path 1>`
	
	`2.repository=<repository code 2>`
	`2.class=<class full path 2>;<class full path 3>;...`
	`;<class full path n>`
	
	`3.repository=<repository code 3>`
	`3.class=*`

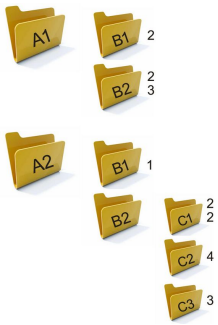
Table 4.3. Examples of configuration to filter on classes

Note

When you choose `n.class=*`, BatchReporting processes all *leaf* classes.

Note

When you explicitly specify a class, BatchReporting processes this class. If this specified class is a node having sub classes, the reported value is the total of all these sub classes. For example with the following class tree:



Configuration	Total returned by BatchReporting
`n.class=A1/B2`	5
`n.class=A2`	12
`n.class=A2/B2`	11

2.3. The types of request

BatchReporting can generate usage report for all the request types. A single type or a types list can be selected. The request types in the list are separated by ";". The possible request types are: `archive`; `retrieval`; `synchronous retrieval`; `migration` ; `check`; `deletion`; `recovery`; `copy`; `synchronization`; `postpone`; `media migration`; `recycling`; `archive restitution`.

This selection is specified with *`request.types`* parameter.

To process with all request types: *`request.types`*=*

2.4. The time period

The last criterion concerns the date range. BatchReporting considers only the requests that are processed during a specific date rage, that you select by setting *`start.date`* and *`end.date`* parameters.

3. Customizing the report

The report is in the form of a CSV file. This file is created in *`output.dir`* directory. BatchReporting creates this directory when it does not exists.

BatchReporting allows you to customize the content and formatting of the CSV file. This customization is done by parameters, which are specified in the main configuration file of BatchReporting.

To generate the report, BatchReporting considers the selected requests in a particular time range (see page 7, «Filtering the report » for more details). The selected requests are grouped by `day` or `month`, depending on the *`group.by`* parameter's value in the general configuration file. A line is generated in the output for each day or month included in the time period.

Note

When no request has been processed for a given day or month, for none of the entities concerned by the report, BatchReporting displays a corresponding line in the report *only if* *`csv.add.empty.lines`* *is set to* *`true`*. Otherwise, the concerned day or month does not appear in the report.

In addition to listing the number of requests per time period, BatchReporting can also include in the report the total size of the data concerned by these requests. This can be done by setting *`size`* parameter to `true` in the general configuration file. When reporting the total size, BatchReporting uses the value of *`size.data.unit`* parameter to determine the unit of the size columns.

BatchReporting allows you to customize the formatting of the output file:

- By default, BatchReporting adds a header at the beginning of the report. This header contains information about the report. You can disable this header by setting *csv.print.header* parameter to `false` in the general configuration file.

  Below is an example of header added by BatchReporting in the CSV report:

```
App property file name=etc/example.properties
arcsys.httpClientProperties.file=etc/http-client-lat036-postgres.properties
entity.type=repository
request.types=archive
size=true
size.data.unit=KILO_BYTE
1.repository=repository 1
2.repository=repository 2
start.date=2023/01/01 00:00:00
end.date=2023/12/31 23:59:59
group.by=DAY
```

- By default, BatchReporting adds column headers in the CSV report.

  The first column title depends on the value of *group.by* parameter. When set to `day`, the first column is entitled "`DAY`". When set to `month`, the first column is entitled "`MONTH`".

  The title of the other columns consists of two parts. The first part is "`#`" if it is a count column and "`SZ`" if it is a size column. The last part is the name of the entity. For a repository, its name corresponds to its code; for a collection, its name is of the form "`<repository code>/<collection code>`"; for a class, its name is of the form "`<repository code>\<class full path>`".

- You can specify the column separator by setting *csv.separator* parameter.

- You can specify the decimal separator by setting *decimal.separator* parameter.

Note

The decimal numbers are represented with a level of precision of three decimal digits.

- You can specify the format for the date column by using *output.date.format* parameter.

  You can specify a format using the following pattern: `YYYY` or `YY` for the year, `MM` for the month and `DD` for the day. `DD` value is only supported when *group.by* parameter is set to `day`.

4. Automatic resumption of an interrupted report

Following a power outage, network failure, or other reason, the construction of the report may be interrupted. In order to allow the reporting to be resumed without having

to start from the beginning, BatchReporting saves an intermediate status of the CSV file in a sub-directory of the output directory. This directory and sub-directory are configured by the *output.dir* and *output.subdir.temporary.csv* parameters.

This temporary CSV file is associated with another file, containing a description of the current report.

The saving frequency of the temporary CSV snapshot generation is specified in the *temporary.csv.frequency* parameter. This frequency corresponds to a number of ended requests of *request.types* type.

4.1. Algorithm description

If the report ends without any interruption, both temporary files are automatically deleted.

If the report is interrupted before the end of its processing, both temporary files will be used by BatchReporting to resume the report.

4.2. Deactivating the automatic resumption of the report

When restarting BatchReporting, if you do not want an interrupted report to be resumed, you must manually delete both temporary files.

4.3. Resume files location

The temporary files are created in the `<output.dir>/<output.subdir.temporary.csv>/` folder.

The temporary files name is:

- when filtering on repositories or collections:
  - when *size* parameter is set to `true`, the file names are of the form: `<report.type>#<entity.type>[<entity id>]#<request.types(1)>#<request.types(2)>#...<request.types(n)>#withRequestSize#<start.date>#<end.date>`
  - when *size* parameter is set to `false`, the file names are of the form: `<report.type>#<entity.type>[<entity id>]#<request.types(1)>#<request.types(2)>#...<request.types(n)>#withoutRequestSize#<start.date>#<end.date>`
- when filtering on classes:

- when *size* parameter is set to `true`, the file names are of the form: `<report.type>#<entity.type>[<repository id>-<class full path>]#<request.types(1)>#<request.types(2)>#...<request.types(n)>#withRequestSize#<start.date>#<end.date>`
- when *size* parameter is set to `false`, the file names are of the form: `<report.type>#<entity.type>[<repository id>-<class full path>]#<request.types(1)>#<request.types(2)>#...<request.types(n)>#withoutRequestSize#<start.date>#<end.date>`

5. Module description

The BatchReporting module is installed in a directory called `batch-reporting`.

1. Directory structure

Type	Name	Description
	`batch-reporting.sh` (or .cmd)	BatchReporting launch script.
	`bin`	Directory containing the executables for the installation platform.
	`etc`	Directory containing the module configuration files.
	`<global_configuration_file>.properties`	File containing the global parameters.
	`http-client.properties`	File containing the parameters of the logs.
	`log4j2.xml`	File containing the parameters of the Arcsys REST API.
	`lib`	Directory containing the Java libraries (JAR) for BatchReporting.

Table 5.1. Content of the `batch-reporting` directory

2. etc/<global_configuration_file>.properties file

This file contains the main configuration of module. In this file, you specify the various criteria used to filter the report and you customize the result file.

Parameter name	**report.type**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	This parameter specifies the type of report. **Note** When *report.type*=`count-size requests ended`, only requests in ended status are considered by BatchReporting. When *report.type*=`count requests errors`, only requests in error status or aborted status are considered by BatchReporting.
Possible values	`count-size requests ended`, `count requests errors`
Shipped value	`count-size requests ended`

Parameter name	**request.types**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	This parameter specifies the types of requests that are concerned by the report.
Possible values	`archive`; `synchronous retrieval`; `retrieval`; `migration`; `check`; `deletion`; `recovery`; `copy`; `synchronization`; `postpone`; `media migration`; `recycling`; `archive restitution` *
Shipped value	`archive`

Parameter name	**entity.type**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	This parameter specifies the type of Arcsys entity concerned by the report. The *entity.type* parameter must be followed by the specification of the code of the entities that are concerned by the report (see page 7, « Filtering the report » for more details).
Possible values	`repository`, `collection`, `class`
Shipped value	`repository`

Parameter name	**size**
Parameter type	Boolean which can be either `true` or `false`.
Parameter change assessment	After module reboot.
Description	This parameter defines if the total size is also computed in addition to the counting operation. **Note** Setting this parameter to `true` will reduce performance.
Shipped value	`true`

Parameter name	**start.date**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	Start date. The expected pattern is `YYYY/MM/DD HH:MM:SS`.
Shipped value	`2020/01/01 00:00:00`

Parameter name	**end.date**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	End date. The expected pattern is `YYYY/MM/DD HH:MM:SS`.
Shipped value	`2020/12/31 23:59:59`

Parameter name	**group.by**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	Grouping unit (see page 11, « Customizing the report » for more details).
Possible values	`day`, `month`
Shipped value	`month`

Parameter name	**output.date.format**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	This parameter allows you to set the date format for the first column of the CSV report (see page 11, « Customizing the report » for more details).
Shipped value	`YY/MM`

Parameter name	**size.data.unit**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	This property allows you to set the size data unit.
Possible values	`Kb`, `Mb`, `Gb`, `Tb`
Shipped value	`Kb`

Parameter name	**output.dir**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	This parameter is used to specify the CSV output directory, where the report is created.
Shipped value	`reports`

Parameter name	**output.subdir.temporary.csv**
Parameter type	Characters string.
Parameter change assessment	After module reboot.
Description	This parameter is used to specify the output sub-directory for the temporary CSV snapshot generation. The value specified corresponds to a relative path, that is a subdirectory of the directory specified by the *`output.dir`* parameter. The directory used for temporary files is therefore: `<output.dir>/<output.subdir.temporary.csv>`. If the sub-directory set does not exists, BatchReporting creates it.
Shipped value	`temp`

Parameter name	**temporary.csv.frequency**
Parameter type	Positive integer.
Parameter change assessment	After module reboot.
Description	This parameter is used to specify the saving frequency of the temporary CSV snapshot generation. This frequency corresponds to a number of ended requests of *`request.types`* type.
Shipped value	`10000`

Parameter name	**csv.separator**
Parameter type	Character.
Parameter change assessment	After module reboot.
Description	This parameter defines the character used to separate the fields of a line in the CSV file. This character must be different from the one set for *`decimal.separator`* parameter.
Shipped value	`;`

Parameter name	**decimal.separator**
Parameter type	Character.
Parameter change assessment	After module reboot.
Description	This parameter defines the decimal separator in the CSV file. This character must be different from the one set for *`csv.separator`* parameter.
Shipped value	.

Parameter name	**csv.print.columns.titles**
Parameter type	Boolean which can be either `true` or `false`.
Parameter change assessment	After module reboot.
Description	This parameter defines if the titles of the columns will be inserted in the CSV file.
Shipped value	`true`

Parameter name	**csv.print.header**
Parameter type	Boolean which can be either `true` or `false`.
Parameter change assessment	After module reboot.
Description	This parameter defines if BatchReporting header will be inserted in the CSV file.
Shipped value	`true`

Parameter name	**csv.add.empty.lines**
Parameter type	Boolean which can be either `true` or `false`.
Parameter change assessment	After module reboot.
Description	This parameter defines if the lines without any activity are nonetheless reported in the CSV file.
Shipped value	`false`

3. etc/http-client.properties file

3.1. HTTP connection settings

Arcsys uses the HTTP(S) protocol to connect to the Arcsys REST API. The following elements of the "`http-client.properties`" file define the connection to this API.

Property name	Description	Default value
`base.url`	Base URL, formed like: `<protocol>://<domain_name or ip_address>[:<port_number>][/<uri>]` `protocol` may have the value `http` or `https`. SSL will automatically be used when the protocol is https.	

Property name	Description	Default value
connect.timeout.ms	This parameter allows to determine the timeout in milliseconds until a connection is established. For infinite timeout, set this parameter to `0`. A negative value is interpreted as undefined (system default).	`1000`
request.timeout.ms	This parameter allows to determine the timeout in milliseconds used when requesting a connection from the connection manager. For infinite timeout, set this parameter to `0`. A negative value is interpreted as undefined (system default).	`1000`
socket.timeout.ms	This parameter allows to define the socket timeout in milliseconds, which is the timeout for waiting for data or, put differently, a maximum period inactivity between two consecutive data packets. For infinite timeout, set this parameter to `0`. A negative value is interpreted as undefined (system default).	`300000`
response.page.size	This parameter corresponds to the page size of the HTTP response for requests getting list of entities. **Note** The higher the value is, the fewer HTTP calls will be, but the higher the value of *request.timeout.ms* delay must be set (as the messages sent from the server to the client are heavier).	`1000`
oauth.token.url	URL to get a OAuth2 token. URL, formed like: `<protocol>://<domain_name or ip_address>[:<port_number>][/<uri>]`	
oauth.authorization.grant.type	This parameter allows to configure the OAuth2 authorization grant type. Possible values are: `authorization-code`, `implicit`, `password`, `client-credentials`	`password`

Property name	Description	Default value
oauth.client.id	This parameter allows to configure the OAuth2 client identifier.	
oauth.client.secret	This parameter allows to configure the OAuth2 client secret password. The value must be an encrypted characters string. To encrypt a clear characters string, use the `encrypt` encryption utility provided in the Arcsys installer. The encrypted characters string will be surrounded by `ENC(...)`.	
oauth.user.name	This parameter allows to configure the OAuth2 user name.	
oauth.user.password	This parameter allows to configure the OAuth2 user password. The value must be an encrypted characters string. To encrypt a clear characters string, use the `encrypt` encryption utility provided in the Arcsys installer. The encrypted characters string will be surrounded by `ENC(...)`.	
ssl.truststore.file	This parameter allows to configure the full path of the trust store file. In SSL mode this parameter is mandatory.	
ssl.truststore.type	This parameter allows to configure the type of trust store. Possible values are: `jks`, `pkcs12`.	`jks`
ssl.truststore.password	This parameter allows to configure the password of the trust store file. The value must be an encrypted characters string. To encrypt a clear characters string, use the `encrypt` encryption utility provided in the Arcsys installer. The encrypted characters string will be surrounded by `ENC(...)`.	
ssl.keystore.file	This parameter allows to configure the full path of the key store file. This parameter is only considered in SSL mode and is optional.	
ssl.keystore.type	This parameter allows to configure the type of key store. Possible values are: `jks`, `pkcs12`.	`jks`
ssl.keystore.password	This parameter allows to configure the password of the key store file. The value must be an encrypted characters string. To encrypt a clear	

Property name	Description	Default value
	characters string, use the `encrypt` encryption utility provided in the Arcsys installer. The encrypted characters string will be surrounded by `ENC(...)`.	

4. etc/log4j2.xml file

This file contains the settings for the log files. The logs are generated with the Apache library log4j.

The parameters of this file depend on the information desired in the logs.
For any changes, consult the standard documentation of Log4J. See: https://logging.apache.org/log4j/2.x/

Glossary

API *(Application Programming Interface)*
The APIs supplied by Arcsys are used by the product holder to completely customize a new Application or User interface to meet the precise ergonomic needs of a given application. A distinction must be made between:

- low-level APIs available in Arcsys Core, in RMI or SOAP protocol, providing access to most operations performed by Arcsys (administration, operation, archiving, searches, archive retrieval);
- and high-level APIs, available in the ArcBWS option. Web services in REST protocol provide access to a number of the most-frequently used methods.

Application agent
There are two different types of agents at archiving level: application interface agents and user interface agents. An **application agent** can archive all the objects specific to an application (files, RDBMS table records, etc.), whereas a **web agent** performs both administration functions and manual archiving functions initiated by the user.

Archive Retrieval
Archive retrieval is an operation that makes a copy of a record available to a record requester. This term takes precedence over the term *restore*, which has another meaning at archiving level (restore in the sense of handing back the documents to the organization that created them or to its representatives, then destroying them). Archive retrieval can be complete (misleadingly called a "complete retrieval") or partial (*Partial Archive Retrieval*, misleadingly called a "partial retrieval").
See Also **Archive Restitution**.

Archive Restitution
Archive restitution is the return and transfer of archived documents to their originator, or to a duly appointed person or organization. An Archive Restitution is in Arcsys an Archive Retrieval operation that ends with a Destruction. An Archive restitution operation can only be created through the appropriate operation in the REST API, or by using ArcEP module.
See Also **Archive Retrieval**, **Destruction**.

Arcsys
ERM published by Infotel. Arcsys refers to both the Arcsys Core product and all of its connectors and options.

Arcsys Connector
An Arcsys connector is an operational module generally requiring an additional license used to interface with an external software package (ECM, ERP, Mail) for archiving and/or archive retrieval to and from Arcsys.

Arcsys Core

The Arcsys Core represents all "essential" Arcsys modules, that is Arcsys Database, the Arcsys RMI, TCP/IP and SOAP API, the Arcsys Transfer Server, the Arcsys Engine, the Arcsys Application Agent, the Arcsys Transfer Service, the Arcsys Web Agent, the Arcsys Auto-Archive Agent, the ArcFF format control module.
See Also **Arcsys**.

Arcsys Engine

The central archiving platform on which synchronous and asynchronous archiving, indexing and retrieval processes operate. The engine can spread threads over multiple processors. This guarantees dialogue and traceability between the agents that are associated to it.

Arcsys Option

Arcsys options are added to the Arcsys Core for additional functionalities. They do not necessarily require an additional architectural module. They may be subject to a separate license. The main options are:

- ArcAFP Option (AFP format management)
- ArcBWS (REST web services)
- ArcMover Tape Option (media manager managing file systems and robots in SCSI or ACSLS)
- ArcIP (record ingestion)
- ArcEP (record extractor)
- ArcPAK Option (record compression on ArcMover and native ingestion of compressed files)
- ArcRFT Option (full text search)
- ArcSIGN Option (internal digital signature generation) and ArcVERIF (external digital signature verification)
- ArcCrypt Option (encryption of data at rest)
- ArcCFN (digital vault)
- ArcREF Option (record ingestion by reference)

Attestation policy

An attestation policy allows to define which type of attestation can be generated as well as a set of parameters concerning their generation.

Classification Scheme

A classification scheme represents a hierarchical classification of activities in an organization used to classify archived objects and link them in this way to their creation context. A

classification scheme is made up of categories organized in a hierarchy over a number of levels.

Collection

Set of rules that a record must comply with. The collection is defined via the Web agent or Arcsys API, and comprises information contained in the relational database tables. A collection always refers to two rules: one concerning the **storage policy** and one relating to the **indexing mask**. A collection is assigned to the record on the initial record request.
See Also **Storage policy**, **Indexing mask***(or metadata template)*.

Deletion

MOREQ2010 provides the following definition for this concept: the act of deleting data from the relational database so that no trace remains. Generally speaking, an entity can only be deleted if is not used in a stored record. Otherwise, it can only be destroyed and not deleted, thus leaving a residual entity.
See Also **Destruction**.

Destruction

This is an irreversible action that deletes the documents by applying disposal criteria. It can be associated with the retention of residual information in the relational database.

Disposal

This is the outcome of archived documents when the retention period ends, i.e. generally, destruction or transfer.
See Also **Destruction**, **Transfer**.

The scheduled end of retention date, retention end date

This is the date at which the last retention date is reached. This is a concept used notably in MOREQ2010.

Disposal Hold

Arcsys can be used to "hold" one or more lots archived in the application. This puts a "hold" on the status of the lots by preventing any operation that could modify this status. It freezes the life cycle and prevents any modification on the archive. Furthermore, the user is guaranteed that when the disposal hold is active, the lot will remain in the same state as it was at the moment it was put on hold.

Electronic Attestation

Document produced to attest that an action or an electronic transaction has occurred.

Envelope

Arcsys groups documents stored in the system in envelopes, either created by Arcsys during the archiving process (in this case, files in TAR format), or created prior to Arcsys processing by the user or third-party processes (*native envelopes* in AFP or ZIP format, for example). The representation of an envelope in the Arcsys Database is called a **logical envelope**. Its technical implementation is also called *MoverReference*. Last but not least,

the representation of physical information stored in the envelope in the optional ArcMover module is called *MoverMedia*.

Event

In Arcsys, a retention schedule can associate the start of record retention with an event with a known or unknown date. This event, created in an Arcsys repository, can thus be attached to a number of different retention schedules.
See Also **Retention schedule**.

Export

An export procedure is used to retrieve the ERM object with a view to transferring to another system and if appropriate deleting it from the ERM. The export must include the object, its information stored in the relational database, its traces and ideally its management rules.

Format policy

A format policy is used to define a set of rules concerning format checks for a given file type. These rules are used to specify the action that will be performed, the expected results of these actions, as well as the error cases, triggering archiving failure.

Hash value

Also called an "integrity certificate" in cryptography, this is the digest of a message which guarantees a practically unique result that is impossible to reverse calculate. The most commonly used algorithms are MD5 (128 bits), SHA-1 (160 bits), SHA256 (256 bits) and SHA512 (512 bits). Arcsys includes a module that is capable of dynamically calling several algorithms. The choice of an algorithm type is valid for all archived objects within the same Arcsys product version; compatibility with algorithms from the previous version is guaranteed. The associated term *hash function* is also used.

Indexing mask*(or metadata template)*

As is the case with the storage policy, an indexing mask is a rule that is referenced by a collection. An indexing mask can be referenced by several collections. An indexing mask refers to the use of a set of Keyword = Value pairs. The keyword component is set to make sense in a specific business application (e.g. Accounting Day, Department, Account No., Account Holder, etc.). The value component can be either unrestricted, or restricted to a set of acceptable values (e.g. A, B or C), or in date format, or restricted by an input mask. Some pairs are defined as mandatory whereas others may be optional.

An application which uses an indexing mask through a collection must supply all Keyword=Value pairs as they are defined using this mask. Any indexing-related errors lead to the record being rejected for conformity. This record is then added to the list of records with errors.

The indexing mask is defined by an administrator via the Arcsys interface or APIs. It is comprised of a set of metadata element definitions.

Fixity

The quality of a document that has not been subject to intentional or accidental destruction, alteration or modification.

Lot

Arcsys can consolidate several different objects that form a functional set in a client application in the same physical record. It is comprised of different types of objects: files, databases, or any other type of object managed by Arcsys. It is possible to retrieve the entire lot or one of the objects contained in the lot. The MOREQ2010 record is translated in Arcsys implementation by a lot; the lot, as opposed to a MOREQ2010 record, can represent documents that are not yet archived.

Manifest

The manifest is an XML file that defines precisely the content of a record. The manifest contains: metadata associated with the record, structure metadata, a description of the physical files of the record(s) that follow, the object-by-object content of the record, object formats, object names, their size, hash value, the algorithm used to calculate the hash value, etc. This is a type of complete ID card for the record.

The manifest is always written on the storage media and precedes the record that it describes. This process is used to automatically describe storage media (irrespective of the medium). With this system, users can understand media content and metadata without installing the software that generated the records.

Metadata element definition (or keyword)

Component of an indexing mask. We use the term "metadata element definition" rather than the term "keyword" as it is closer to MOREQ2010. The metadata element definition in particular defines the type of metadata (date, string, digital, controlled) and its input mask, for example.
See Also **Indexing mask***(or metadata template)*.

MOREQ2010

MOREQ2010 (Modular Requirements for Records Systems) is a standard made available by the European Commission (DLM Forum Foundation) for electronic archiving. As Arcsys pre-dates MOREQ2010, it is taken into account in Arcsys Core via an add-on that is used to progressively include MOREQ2010 entities (aggregate, classification scheme, etc.), on top of the Arcsys entities (repository, collection, lot).

Nearline

Storage level managed through a media manager (XBSA interface). This storage level can be used with or without online storage. It creates copies on the storage media supported by the chosen media manager (disks, tapes, optical disks, etc.). This storage level is optional. All types of online storage, nearline disks or nearline cartridges support periodical content verification functions, continuity of the formats used and the portability of fixity when technical or planned migrations are performed via the collections.

Object

The object is a basic archived unit that can be retrieved via Arcsys. Lots contain one or more objects. An object can be: a file, a directory, a table, a relational table, etc. The MOREQ2010 component is implemented by this object concept; the object, as opposed to a MOREQ2010 component, can represent a document that has not yet been archived.

Online
Storage level, which must be disk type, that makes records permanently available within an extremely reduced time period. Arcsys also allows use of a Nearline storage level (see definition).

Permissions
Permissions refer to the user profiles or groups authorized to access documents or sets of documents archived in the system.

Proof folder
This is comprised of a record, a proof slip, and additional items where required (functional traces, for example) that are used, by demonstrating the fixity and the authenticity of a document, for admission as proof.

Record
A record is an evidential document that is deemed sufficiently important by the creator to be managed by an ERM that will manage its life cycle (retention, disposal, etc.). A record represents an archived lot. A record is archived via a *record request*. Archiving a document *creates a record*.

Relational database *(previously called "referential")*
Essential component of the system, it contains all the data (excluding archived data) used by Arcsys for its operation. It includes logical entities called "repositories" (see definition).

Repository
This is a logical entity in the Arcsys Database. The company can define as many repositories as it wants, either to define a test set, to isolate an application, or for any other reason. These repositories are entirely independent of each other. They all have their own pattern and all have the same structure.

Restore*(correct term: retrieval)*
This term is used misleadingly in Arcsys to refer to the concept of archive retrieval. It is not accepted in archiving terminology as to mean transfer and then destruction.
See Also **Archive Retrieval**.

Retention and disposal schedule
This comprises all the rules defining the record retention period for a company or an organization, according to risks of unavailability and information system access requirements. It specifies the disposal after these time periods.
See Also **Retention schedule**.

Retention period
A duration expressed in days, months or years of object retention. This is a concept used notably in MOREQ2010.

Retention schedule
A retention schedule defines the start and the end of the retention of records that are attached to it, either directly or through their class.

Retention start date
Date from which a retention period must be taken into account. This is a concept used notably in MOREQ2010.

Security
An ERMS requirement that involves including documents whose use (confidentiality, risk of exposure) and/or fixity (non modification of content, non-alteration of media) should be closely monitored.

Storage pool
The storage policy assigns a "zone" to a "policy". This is a logical storage pool, characterized in particular by its time period (e.g. 10 years).

Storage policy
A storage policy is a rule that is referenced by a collection. The policy dictates the storage media which are successively implemented to hold a record, as well as the retention period for each media. The storage policy is defined through the graphical interface. Applications or business users use it indirectly through the reference to a collection. A storage policy can be changed over time to reflect new retention periods or new storage media. The policy covers storage units by logical pool.

Storage zone
The storage zone is a logical entity representing a physical storage space (e.g. set of file systems, robot, zone NetBackup, TSM, Cloud or ArcMover).

Synchronous retrieval
This is an archive retrieval that takes place in the form of a direct retrieval of a document (for direct viewing or downloading) in a Web browser. This term takes priority over the term "online viewing" or, misleadingly, "online restore".
See Also **Archive Retrieval**.

Time stamping
Time stamping is a technique used to associate a document with a certain date in reference to a given and recognized time system. The date set in this way is an essential element for document authentication. Time stamping can be performed internally or by a third-party time stamp.

Tracking
This is the result of continuously creating, capturing and maintaining information about the movement and use of the system and objects (ISO 15489-1:2001, 3.19).

Transfer
In an archival sense, this operation sends an archived object to another IT system. Once the transfer is performed, the object can be removed from the ERMS as needed. In OAIS terminology, a transfer represents more specifically the physical transmission of a record or a set of records by a service supplying an archive service. Not to be confused with the transfer of data in the purely technical sense, as with the Arcsys Transfer Server module.

Registered Trademarks

Firefox is a registered trademark of the Mozilla Foundation.

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of the Open Group.

Microsoft Windows, Windows NT, Windows Server, SQL Server, Internet Explorer are registered trademarks of Microsoft Corporation in the United States and/or other countries.

SAP is a registered trademark of SAP AG in Germany and other countries.

MySQL is a registered trademark of Oracle and/or its subsidiaries. MariaDB is a registered trademark of Monty Program Ab.

Java is a registered trademark of Oracle and/or its subsidiaries in the United States and other countries.

Infotel is a registered trademark of Infotel SA.

All trademarks mentioned are the property of their respective owners.

Infotel Technical Support

https://techsupport.infotel.com