



# Arcsys - ArcREF Option

## ArcREF Option Guide

25.3.2.STS  
March 23, 2026

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

Version: 25.3.2.STS

Publication date of the document: 2026-03-23

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:**

<b>France</b>	<b>Germany</b>	<b>USA</b>
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
<a href="https://techsupport.infotel.com">https://techsupport.infotel.com</a> <software@infotel.com>	<a href="https://techsupport.insoft-software.com">https://techsupport.insoft-software.com</a> <software@insoft-infotel.com>	<a href="https://techsupport.infotel-consulting.co.uk">https://techsupport.infotel-consulting.co.uk</a> <software@infotel.com>

Copy prohibited without explicit authorization. © 2026 Infotel SA All rights reserved.

## Table of Contents

Preface .....	6
1. Introduction .....	6
2. Reference Documents .....	6
2.1. Concepts .....	6
2.2. Installing and Updating .....	6
2.3. Operations .....	6
2.4. GUI .....	6
2.5. Development .....	6
2.6. Option guides .....	6
2.7. Optional modules .....	7
3. Symbols and Meanings .....	7
4. Definitions and Abbreviations .....	7
1. Principles .....	1
1.1. Archiving by reference .....	1
1.2. A batch utility to inject archives by reference: ArcREF Injector .....	1
1.3. Compatibility limitations .....	2
1.3.1. Architecture and consequences .....	2
1.3.2. Platforms and options .....	2
1.3.3. Tracability .....	2
1.3.4. Storage profile and retention schedule characteristics .....	2
1.3.5. Search and metadata .....	3
1.4. Content of the storage .....	3
1.5. Access to the referenced files .....	3
2. Installing and configuring ArcREF Injector .....	4
2.1. Prerequisites .....	4
2.2. Installing the module .....	4
2.3. Finalizing installation .....	4
2.3.1. database-settings.conf .....	4
2.3.2. arcrefinjector.conf .....	5
3. Configuring archive by reference injection .....	8
3.1. Zones configuration .....	8
3.1.1. Filesystem storage configuration .....	8
3.1.2. Cloud storage configuration .....	9
3.2. Writing a settings file .....	9
3.2.1. Content .....	9
3.2.2. Detailed example .....	10
3.2.3. Sample .....	10
3.3. Writing the SIP XML file(s) .....	10
3.3.1. Structure of the SIP XML file .....	11
3.3.2. Sample .....	16
4. Running ArcREF Injector .....	17
4.1. Command line .....	17
4.2. Result of the execution .....	18

4.3. Reentrance of the execution .....	19
4.4. When the injection is done .....	19
4.5. Identifying archives by reference .....	19
Glossary .....	20
Registered Trademarks .....	28

## List of Tables

2.1. Parameters of database-settings.conf .....	4
3.1. SIP V3.1 elements at "Archives" and "Archive" level .....	11
3.2. SIP V3.1 elements at "Document" level .....	12
3.3. Link between size and parameters .....	14
3.4. Link between checksum and parameters .....	15

# Preface

## 1. Introduction

This document supplies detailed information for the use of ArcREF Option of the Arcsys product.

## 2. Reference Documents

### 2.1. Concepts

Arcsys Presentation Manual: **Arcsys-presentation-25.3.2.STS-en.pdf**

Arcsys Functional Description Manual: **Arcsys-functional-description-25.3.2.STS-en.pdf**

### 2.2. Installing and Updating

Arcsys Prerequisites Manual: **Arcsys-requirements-25.3.2.STS-en.pdf**

Arcsys Installation Manual: **Arcsys-installation-25.3.2.STS-en.pdf**

### 2.3. Operations

Arcsys Administration Manual: **Arcsys-administration-25.3.2.STS-en.pdf**

Arcsys Errors Manual: **Arcsys-error-25.3.2.STS-en.pdf**

### 2.4. GUI

Arcsys Web Interface User Manual: **Arcsys-web-25.3.2.STS-en.pdf**

Interface Guide: **Arcsys-web-end-user-25.3.2.STS-en.pdf**

### 2.5. Development

Arcsys API Manual: **Arcsys-api-25.3.2.STS-en.pdf**

### 2.6. Option guides

ArcHP Option Guide: **Arcsys-option-archp-25.3.2.STS-en.pdf**

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

ArcREF Option Guide: **Arcsys-option-arceref-25.3.2.STS-en.pdf**

## 2.7. Optional modules

BatchReporting: **BatchReporting-UserGuide-25.3.2.STS-en.pdf**

ClassAssigner: **ClassAssigner-UserGuide-25.3.2.STS-en.pdf**

MetadataReplacement: **MetadataReplacement-UserGuide-25.3.2.STS-en.pdf**

StartRetentionDateAssigner: **StartRetentionDateAssigner-UserGuide-25.3.2.STS-en.pdf**

## 3. Symbols and Meanings



### Note

Identifies information of particular interest



### Important

Identifies important information

## 4. Definitions and Abbreviations

See the [Glossary](#)

# 1. Principles

## 1.1. Archiving by reference

ArcREF Option is an option of Arcsys that accelerates decommissioning operations by eliminating the need for data migration and the typical procedures, which include:

- Extracting archives from the previous EDRMS
- Injecting via ArcIP
- Managing work disk space
- Dealing with complex scheduling

With the "archiving by reference" approach, you only need to extract the technical and business metadata and the reference to the existing archives (path and filename) to create new archives.

The existing files will be treated as envelopes of a type known as "RAW".

Archives of "Raw files" can be ingested by Arcsys through the ArcREF Injector module, which employs the same mechanism as ArcIP with sip.xml deposits.

## 1.2. A batch utility to inject archives by reference: ArcREF Injector

ArcREF Injector is included in the standard packaging of Arcsys.

It enables the injection of technical and business metadata through an advanced SIP based on that of ArcIP, which includes, in addition to the "standard" SIP:

- At the archive level: repository, collection, retention schedule, permissions...
- At the object level: file attributes (size, Unix rights, etc.), location of the file (filesystem and path) for each copy, file name...

This advanced SIP handles multiple archives per SIP and multiple objects by archive.

ArcREF Injector also provides optional control over file attributes, which requires access to the data.

For performance reasons, by default, hash values verification is disabled. In this case, the hash values verification is only done by Arcsys when accessing the data. If you want ArcREF Injector to verify the hash value during the injection, it can be activated by using `object.stat.hash.verification` parameter.



## Note

Hash values verification is only available for archives stored on filesystem. It cannot be used for Cloud S3 storage.

## 1.3. Compatibility limitations

### 1.3.1. Architecture and consequences

For performance reasons, ArcREF Injector is a binary module that directly inserts data into the database through the Arcsys TCP/IP API.

As a result, it does not provide access to all the capabilities of Arcsys available through Arcsys REST API.

### 1.3.2. Platforms and options

ArcREF Injector is compatible only with the following platforms:

- OS: Linux, Windows
- Database: all databases are supported
- Media types: only ArcMover Disk and cloud S3 storage (with ArcMOVS3 Option) are supported
- Compatibility with ArcPAK Option and ArcAFP Option: AFP and ZIP files are not supported (to be considered as native files). The data cannot be provided already compressed on the media. However, by using the mechanism of storage pool migration, it is possible to migrate the data to a pool compressed with ArcPAK Option.
- Compatibility with ArcCrypt Option: as for compression, already encrypted files are not supported, but encryption can be added afterwards and activated through the migration of the archives.

### 1.3.3. Tracability

Traces by collection and XML Premis Events are not generated by the injection.

### 1.3.4. Storage profile and retention schedule characteristics

The storage profile can maintain multiple copies. The zone targeted for reference-based injection may have copies that are either also subject to reference-based

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

injection or automatically trigger the creation of additional copies in those zones. Additionally, the zone targeted for reference-based injection can be followed by other zones, which will later trigger migrations to those zones.

All types of retention schedules are supported.

### 1.3.5. Search and metadata

All types of keywords are supported, except the extended date subtype of string metadata.

For dates keywords, the \$TODAY value is supported, but not \$1MONTH.

ArcRFT Option and GenericSearch are not supported.

## 1.4. Content of the storage

Each referenced file is created as a native object (of type: RAW) which is also an envelope in Arcsys. The original filename is never modified during the archiving operation.

During migration, recovery, or copying, the new envelope created is renamed on the target zone to follow the Arcsys naming pattern "lot\_nnn\_seq\_x", but the content of the file remains identical.

No XML manifest is created until a synchronization is triggered (the sequence number starts with 2).



### **Important**

**FsComparator is not compatible with RAW objects and cannot be used as long as the files have not been migrated to another storage zone.**

## 1.5. Access to the referenced files

It is not mandatory to provide access to the physical storage where the files are located to the injector.

Physical access is only necessary if verification of the physical characteristics (size, UID, etc.) is requested.

Please refer to the [page 9](#), « Writing a settings file » chapter for more details.

## 2. Installing and configuring ArcREF Injector

### 2.1. Prerequisites

The prerequisites for the module align with the prerequisites of Arcsys as detailed in [Arcsys Prerequisites Manual](#).

### 2.2. Installing the module

To install the module, please refer to the [Arcsys Installation Manual](#).

### 2.3. Finalizing installation

Before running ArcREF Injector, you must define the values in the `database-settings.conf` configuration file for database access. Additionally, review the values in the `arcrefinjector.conf` configuration file for specific injection parameters.

#### 2.3.1. database-settings.conf

This file allows you to configure the connection to the Arcsys TCP/IP API module for referencing archives in the database.

Except for the `arcsys.db_type` parameter, which can only have the possible value `apitcp` (the parameter is defined for future use), all other parameters in this file are equivalent to the parameters in the `transferServer.conf` configuration file of the Arcsys Transfer Server, as described in the [Arcsys Administration Manual](#) (see [Arcsys Administration Manual](#)). Here is a table showing the matching between the parameters.

Parameter name	Matching transferServer.conf parameter
<code>arcsys.api.host</code>	<code>api_host_name</code>
<code>arcsys.api.port</code>	<code>api_port_number</code>
<code>arcsys.pool.connections</code>	<code>database_pool_connections_size</code>
<code>arcsys.pool.connection.max</code>	<code>database_max_pool_connections_size</code>
<code>arcsys.pool.connection.refresh</code>	<code>database_refresh_connections</code>
<code>security_protocol</code>	<code>security_protocol</code>
<code>ssl_keystore_filename</code>	<code>ssl_keystore_filename</code>
<code>ssl_keystore_password</code>	<code>ssl_keystore_password</code>
<code>ssl_truststore_filename</code>	<code>ssl_truststore_filename</code>
<code>ssl_truststore_password</code>	<code>ssl_truststore_password</code>
<code>ssl_library_libssl</code>	<code>ssl_library_libssl</code>
<code>ssl_library_libcrypto</code>	<code>ssl_library_libcrypto</code>

<code>ssl_session_timeout</code>	<code>ssl_session_timeout</code>
----------------------------------	----------------------------------

**Table 2.1. Parameters of database-settings.conf**

## 2.3.2. arcdefinjector.conf

- `injection.errors.max`: the number of errors allowed during referencing, after which the process is stopped. Default : 100
- `injection.threads`: the number of simultaneous threads processing the referencing operation. Increasing the value will require more simultaneous database connections. Default: 2
- `injection.simulation.mode`: set it to `true` to simulate the injection without inserting anything into the database (the SIP file is checked with the XSD). Default: `false`
- `xsd.validation`: set it to `true` to validate the XML files (SIP and settings file). Default: `true`
- `xsd.archive.filename`: the path of the XSD file used to validate the SIP XML file. Default: `xml/sip-3.1.xsd`
- `xsd.arcsys.configuration.filename`: the path of the XSD file used to validate the settings file. Default: `xml/arcsys-settings-1.0.xsd`
- `object.validation.unicity`: this parameter verifies the uniqueness of file paths in the SIP XML file. Setting it to `false` (if you are sure that the file paths are unique) can enhance performance in a database with millions of rows for the documents. Default: `true`. Adding an index on the document path can improve performance for large datasets. An example of creating such an index on Oracle is:

```
CREATE INDEX AR0005_OBJECT_OBJNAME ON AR0005_OBJECT (OBJNAME);
```

- `object.stat.verifcation`: define it to `true` to check the characteristics (UID, GID,...) written for each document in the SIP XML file. Default: `false`.

For the size:

- if the parameter `object.stat.verifcation` is `true` and if the real size computed by the injector is different: if `object.stat.override` is `false`, the injection will stop, if `object.stat.override` is `true`, the real size will override this theoretical size.
- if the size is not specified: if the parameter `object.stat.verifcation` is `true`, it will be automatically computed. Otherwise, the injection will fail.

For the uid and other optional characteristics:

- If the parameter `object.stat.verifcation` is `true` and if the value computed by ArcREF Injector is different from the one set in the SIP XML: if

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

*object.stat.override* is false, the injection will fail, if *object.stat.override* is true, the value computed by ArcREF Injector will override the one set in the SIP XML.

- If the uid is not specified, this will not make the injection fail whatever the value of *object.stat.verification*.
- *object.stat.hash.verification*: define it to true to check the hash value written for each document in the SIP XML file. Default: false. This parameter will only be considered if *object.stat.verification* is set to true.

The hash verification is only available for documents stored on filesystem.

If *object.stat.hash.verification* and *object.stat.verification* are set to true:

- if *object.stat.override* is set to true, ArcREF Injector computes the checksum of the document and overrides the one specified in the SIP XML (**U9**, **U10** and **U11** use cases in the table page 15, « Link between checksum and parameters »).
- if *object.stat.override* is set to false:
  - if <checksum> tag is set in the SIP XML file with a value that matches the one computed by ArcREF Injector, the injection goes on (**U12** use case in the table page 15, « Link between checksum and parameters »).
  - if <checksum> tag is set in the SIP XML file with a value that is different from the one computed by ArcREF Injector, the injection fails with a hash error (**U13** use case in the table page 15, « Link between checksum and parameters »).
  - if <checksum> tag is not set in the SIP XML file (either the tag is defined with an empty value or the tag is missing from the SIP XML file), the injection fails because of missing information (**U14** use case in the table page 15, « Link between checksum and parameters »).
- *object.stat.override*: this parameter is considered only when *object.stat.verification* is set to true. Refer to the description of *object.stat.verification* for more details. Default: false.
- *metadata.string.insertvalue.autolock*: This parameter is considered only when *injection.threads* is greater than 1. Its purpose is to automatically lock the insertion of a new value for string metadata cases. This prevents injection failures that could occur if the same value is inserted simultaneously in multiple threads. This lock slightly reduces performance. If you are certain that the values of the metadata are distinct, you may consider setting it to false. Default: true.
- *metadata.string.insertvalue.autolock.timeout*: This parameter is considered only when *metadata.string.insertvalue.autolock* is set to true. It specifies the timeout

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

(in seconds) for waiting for the release of the lock for string metadata values. If the lock is not released after this timeout, the injection will fail. Default: 10.

- *log.filename*: this parameter specifies the path to the log file. Default: `log/arcrefinjector.log`.
- *log.level*: this parameter defines the log level, which can be one of the following: `debug`, `info`, `warn`, `error`. Default: `info`.

## 3. Configuring archive by reference injection

To perform archive by reference injection, you need to configure the storage settings in Arcsys, define the parameters in the injector's configuration file, and create the necessary SIP XML file(s).

### 3.1. Zones configuration

Archiving by reference allows you to reference files in the following ways:

- On a single storage zone: each file is referenced on one zone and does not have any copies. For example, you have a file named `test.pdf` that is located in `/archives/folder1`.
- On several storage zones simultaneously: each file will be present with the same name on multiple zones. These zones can be of the same type (filesystem, cloud S3) or different. For example, you have a file named `test.pdf` that is present in both `/archives/folder1` and `/archives/folder2`.

Before running ArcREF Injector, you must create the zone(s) in Arcsys and configure them properly.

#### 3.1.1. Filesystem storage configuration

You must create an ArcMover Disk zone and one or several filesystems.

There are two possible scenarios:

- If you are certain that no file on the existing filesystem follows the naming pattern `"lot_nnn_seq_x"` (the pattern used for Arcsys envelopes) and you intend to perform manifest synchronizations, recoveries, and create new archives on this existing filesystem, you can have only one filesystem attached to the zone.
- If you are uncertain whether there are already files with the naming pattern `"lot_nnn_seq_x"` on the filesystem, and/or you wish to perform manifest synchronizations, recoveries, and create new archives on another filesystem, you must create and attach at least one additional filesystem to the zone.

Ensure that each of these filesystems is accessible by the Arcsys Transfer Servers.

The size of the filesystems should be set to 0. The injector progressively increases the size of the filesystem as documents are injected, keeping the remaining space automatically at 0.

Obtain the "id" of the filesystems (visible in the web screens) to define the matching between the files path and the filesystems in the settings file of the injector (refer to the page 9, « Writing a settings file » chapter).

### 3.1.2. Cloud storage configuration

Prior to executing ArcREF Injector, you need to configure the cloud S3 storage zone with attributes that match the bucket where the files are located.



#### Important

**Locking a cloud S3 storage zone is not an option in Arcsys, which means you must ensure that no file follows the pattern "lot\_xxx\_seq\_y" to prevent potential overwriting by the product during new archiving operations.**

## 3.2. Writing a settings file

ArcREF Injector takes a parameter in the form of an XML file called the "settings file" specified in the mandatory *arcsys-settings* parameter.

This file must conform to an XSD located in the `xml` folder of the injector, named `arcsys-settings-1.0.xsd`.

This file is used to configure the link with Arcsys.

### 3.2.1. Content

The file consists of a mandatory first part that includes:

- The code of the Arcsys Application Agent associated with the requests in the database;
- The code of the Arcsys Engine associated with the requests in the database.

In both cases, the Arcsys Application Agent and Arcsys Engine are not involved in the technical process of archiving by reference, which is fully managed by the injector. They are only needed to comply with a standard archiving request.

The file also contains a second part that is only necessary if the files need to be referenced on one or more filesystems.

In this case, you must reference all these filesystems by specifying their identifier and their absolute path.



## Note

There is no need to define anything related to cloud S3 configuration in this settings file because the injector will automatically seek the appropriate bucket from the profile configuration of the collection.

### 3.2.2. Detailed example

```
<?xml version="1.0" encoding="UTF-8"?>
<arcsys version="1.0">
  <applicationAgentCode>ApplicationAgentCode</applicationAgentCode>
  <engineCode>EngineCode</engineCode>

  <filesystems>
    <filesystem id='132'>/opt/Infotel/Arcsys/filesystemes/A/</filesystem>
    <filesystem id='267'>/opt/Infotel/Arcsys/filesystemes/B/</filesystem>
  </filesystems>
</arcsys>
```

In this example:

- The Arcsys Application Agent to use has a code `ApplicationAgentCode`
- The Arcsys Engine to use has a code `EngineCode`
- When in the `sip.xml` files are referenced on the 132 filesystem they can be found in the `/opt/Infotel/Arcsys/filesystemes/A/` directory.
- When in the `sip.xml` files are referenced on the 267 filesystem they can be found in the `/opt/Infotel/Arcsys/filesystemes/B/` directory.

### 3.2.3. Sample

A sample file `arcsys-settings-1.0_sample.xml` is available in the `xml` subfolder of the injector.

This sample is the detailed example provided above.

## 3.3. Writing the SIP XML file(s)

The injection is performed by creating SIP XML files that reference the files and their metadata. A SIP XML file can reference one or several archives, that can be in various repositories and collections. Each archive can contain one or several documents. Each execution of the injector takes one SIP XML file in parameter.

This file must conform to one of the XSD located in the `xml` folder of the injector and named `sip-3.0.xsd` or `sip-3.1.xsd`.



## Important

The SIP XML file is loaded entirely into memory. Use caution when determining the number of archives referenced in a single SIP file to ensure it aligns with the available memory on your server. For example, if you have 1 million archives to inject, you can split the injection into 100 SIP XML files, each containing 10,000 archives.

### 3.3.1. Structure of the SIP XML file

The SIP version 3.0 or 3.1 are supported. They must have the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<sip version="3.1">
  <archives>
    <archive>
      <!--...XML elements at "Archive" level: code,label,metadata,references...-->
      <documents>
        <!--...XML elements at "Document" level: nativeFile,metadata...-->
      </documents>
    </archive>
    <archive>
      ...
    </archive>
  </archives>
</sip>
```

Name	Parent tag	Description	Cardinality
archives	sip	Set a list of archives.	1
archive	archives	Set an archive declaration.	1..n
code	archive	Code of the archive in Arcsys. It must not exceed 64 characters.	1
label	archive	Label of the archive in Arcsys. It must not exceed 250 characters.	1
repositoryCode	archive	Code of the repository of the archive. It must not exceed 64 characters.	1
collectionCode	archive	Code of the collection of the archive. It must not exceed 64 characters.	1
author	archive	Author of the archive. It must not exceed 255 characters.	1
domain	archive	Attribute of the archive tag. The domain can have the value business or technical. The default value is business.	1
creationDate	archive	Creation date of the archive. Set it to NULL if not defined.	0..1
retention	archive	It contains the retentionSchedule and startRetentionDate elements.	0..1

retentionSchedule	retention	Set a specific retention schedule using its name, this implies that the one of the class is overloaded and therefore not used.	0..1
startRetentionDate	retention	The retention date matches the start date of record retention in Arcsys. By shifting this date, you can extend or reduce the storage duration of the transfer. Date format must be YYYY-MM-DD . The default value (if no retention start date is specified) is the transfer date. You can opt to not define a retention start date by specifying the value <i>NULL</i> .	0..1
classificationPath	archive	Path of the classification plan to use.	0..1
metadataSet	archive	Declaration of a set of metadata.	0..1
metadata	metadataSet	Declare a metadata: the keyword is set using the <i>key</i> attribute.	0..n
multi-valued-metadata	metadataSet	Declare a multivalued metadata: its different values are declared using <i>values</i> tag following by the <i>value</i> tag for each metadata to specify.	0..n
permissions	archive	Declare a set of permissions.	0..1
permission	permissions	Declare a permission: its type can be specified using the <i>type</i> attribute. The available types are manage, archive, retrieval, download, view.	0..n
group	permission	Declare the group associated to a permission.	0..n
user	permission	The user name of the user to a permission. It can be declared in any order with the group tag.	0..n
references	archive	Set a list of references zones.	1
reference	references	Declaration of a reference zone. Use the <i>id</i> attribute to specify the identifier of the zone.  This element is used to specify on which zone the files are referenced, among all the zones defined in the storage policy. It allows to add other zones to the storage policy (if you intend to perform copies to a new zone for example) where the files are not referenced.	1..n

**Table 3.1. SIP V3.1 elements at "Archives" and "Archive" level**

Name	Parent tag	Description	Cardinality
documents	archive	Declare a set of documents.	1
document	documents	Declare a document.	0..n
domain	document	Attribute of the document tag. The domain can have the value <i>business</i> or <i>technical</i> . The default value is <i>business</i> .	1
metadata	document	Declare a metadata: the keyword is set using the <i>key</i> attribute. Example : <metadata key="CLIENT">Smith</metadata>	0..n

		<p>For numerical values, only digits are extracted. For example a value 10 231 321 or 10,231,321 will be converted to 10231321.</p> <p>For date values, the date can be in ISO format (YYYY-MM-DD, ex. 2022-01-23), ISO date and hour (YYYY-MM-DDTHH:MM:SS, ex. 2022-01-23T09:31:54), or 'date8' format (ex. 20220123).</p>	
multi-valued-metadata	document	Declare a multivalued metadata: its different values are declared using <i>values</i> tag following by the <i>value</i> tag for each metadata to specify.	0..n
domain	metadata,multi-valued-metadata	Attribute of the metadata, multi-valued-metadata tags. The domain can have the value business or technical. The default value is business.	1
nativeFile	document	Declare a file to be referenced. The type is set using the <i>type</i> attribute. The only value that is supported for archiving by reference is RAW. The tag must therefore be: <nativeFile type="RAW">.	0..1
path	nativeFile	Declare the file name of the native file. This path is relative to the storage used (filesystem or bucket). Examples: test.pdf, folder1/test.txt	1
checksum	nativeFile	<p>This tag specifies the file hash. A "type" attribute specifies the algorithm used. Possible values are: MD5*,SHA-1,SHA-256,SHA-512.</p> <p>If the parameter <i>object.stat.hash.verification</i> is true and if the real checksum computed by the injector is different: if <i>object.stat.override</i> is false (<b>U13</b> use case in the table page 15, « <a href="#">Link between checksum and parameters</a> »), the injection will fail. If <i>object.stat.override</i> is true (<b>U10</b>), the real checksum will override the theoretical checksum.</p> <p>For detailed behavior, please refer to the table page 15, « <a href="#">Link between checksum and parameters</a> ».</p> <p>(*): The MD5 algorithm is not recommended.</p>	0..1
size	nativeFile	<p>This tag specifies the size of the file in bytes. If the parameter <i>object.stat.verification</i> is true and if the real size computed by the injector is different: if <i>object.stat.override</i> is false (<b>U4</b> use case in the table page 14, « <a href="#">Link between size and parameters</a> »), the injection will fail, if <i>object.stat.override</i> is true (<b>U3</b>), the real size will override this theoretical size.</p> <p>If the size is not specified: if the parameter <i>object.stat.verification</i> is true (<b>U7,U8</b>), it will be automatically computed. Otherwise (<b>U6</b>), the injection will fail.</p>	0..1

uid	nativeFile	<p>This optional tag specifies the "uid" attribute of the file in octal format. Example: 787. If the parameter <i>object.stat.verification</i> is true and if the real uid computed by the injector is different: if <i>object.stat.override</i> is false, the injection will fail, if <i>object.stat.override</i> is true, the real uid will override this theoretical uid.</p> <p>Whatever the value of <i>object.stat.verification</i>, if the uid is not specified, the injection will succeed.</p>	0..1
gid	nativeFile	<p>This optional tag specifies the "gid" attribute of the file in octal format. Example: 999. The behaviour is the same as the one explained for the uid tag.</p>	0..1
rights	nativeFile	<p>This optional tag specifies the "rights" attribute of the file in octal format. Example:420. The behaviour is the same as the one explained for the uid tag.</p>	0..1
filesystems	document	<p>In the case of filesystem storage, the list of filesystems on which copies of the file are located, with the same relative file name (path node of the nativeFile). The file will be referenced on all of these filesystems.</p>	0..1
filesystem	filesystems	<p>A filesystem configured in Arcsys. The id of this filesystem must be defined with the id attribute. Example : &lt;filesystem id="132"/&gt;</p>	1..n

**Table 3.2. SIP V3.1 elements at "Document" level**

The following table provides a better understanding of the link between the size defined in the SIP.xml and the parameters *object.stat.verification* and *object.stat.override*:

Use case	object.stat.verification	object.stat.override	Size specified in SIP.xml?	Specified size = computed size?	Expected result
U1	true	true	YES	YES	OK
U2	true	false	YES	YES	OK
U3	true	true	YES	NO	OK with size = computed size
U4	true	false	YES	NO	Error (mismatch)
U5	false	true or false (ignored)	YES	No computed size	OK with size = specified size
U6	false	true or false (ignored)	NO	No computed size	Error (mandatory data)

Use case	object.stat.verification	object.stat.override	Size specified in SIP.xml?	Specified size = computed size?	Expected result
U7	true	true	NO	Not possible since no size is specified	OK with size = computed size
U8	true	false	NO	Not possible since no size is specified	OK with size = computed size

**Table 3.3. Link between size and parameters**

The following table provides a better understanding of the link between the checksum defined in the SIP.xml and the parameters *object.stat.verification*, *object.stat.hash.verification* and *object.stat.override*:

Use case	object.stat.verification	object.stat.hash.verification	object.stat.override	Checksum specified in SIP.xml?	Specified checksum = computed checksum?	Expected result
U9	true	true	true	YES	YES	OK
U10	true	true	true	YES	NO	OK with checksum = computed checksum
U11	true	true	true	NO	Not possible since no checksum is specified	OK with checksum = computed checksum
U12	true	true	false	YES	YES	OK
U13	true	true	false	YES	NO	Error (mismatch)
U14	true	true	false	NO	Not possible since no checksum is specified	Error (mandatory data)
U15	true	false	true	YES	YES	OK with checksum = checksum specified
U16	true	false	true	YES	NO	OK with checksum = checksum provided
U17	true	false	true	NO	No computed checksum	Error (mandatory data)
U18	true	false	false	YES	No computed checksum	OK with checksum =

Use case	object.stat. verification	object.stat. hash. verification	object.stat. override	Checksum specified in SIP.xml?	Specified checksum = computed checksum?	Expected result
						checksum provided
U19	true	false	false	NO	No computed checksum	Error (mandatory data)
U20	false	true or false (ignored)	true or false (ignored)	YES	No computed checksum	OK with checksum = specified checksum
U21	false	true or false (ignored)	true or false (ignored)	NO	No computed checksum	Error (mandatory data)

**Table 3.4. Link between checksum and parameters**

### 3.3.2. Sample

A sample file `sip-3.1_sample.xml` is available in the `xml` subfolder of the injector.

This sample references a single archive with various permissions (view, archive, retrieval, manage). This archive has 5 archive-level metadata. The archive contains two files, `invoice.pdf` and `invoice_detailed.pdf`, each with 6 metadata of various types.

## 4. Running ArcREF Injector

### 4.1. Command line

ArcREF Injector can be run with the following command:

```
./launch.sh (or.bat) --license <name of the license file> --arcsys-settings <name of the settings file> --archives <name of the sip.xml file> <optional parameters>
```

With the following mandatory values:

- name of the license file is the path to the file containing a valid Arcsys license.  
*-L* can be used as a short name in place of *--license*
- name of the settings file is the path to the XML settings file that complies with the *arcsys-settings-1.0.xsd* XSD file (refer to the [page 9](#), « [Writing a settings file](#) » chapter).  
*-S* can be used as a short name in place of *--arcsys-settings*
- name of the sip.xml is the path to the SIP XML file that complies with the *sip-3.0.xsd* or *sip-3.1.xsd* XSD file (refer to the [page 10](#), « [Writing the SIP XML file\(s\)](#) » chapter).  
*-A* can be used as a short name in place of *--archives*

Optional parameters can include:

- *--settings* (or *-c*) <injector settings file>: injector configuration settings (by default, the file is loaded from *etc/arcrefinjector.conf*) (refer to the [page 5](#), « [arcrefinjector.conf](#) » chapter)
- *--db-settings* (or *-d*) <database settings file>: Arcsys database configuration settings (by default the file is loaded from *etc/database-settings.conf* (refer to the [page 4](#), « [database-settings.conf](#) » chapter)
- *--csv-errors* (or *-e*) <csv file name>: generates a CSV file for errors (see [page 18](#), « [Result of the execution](#) » chapter)
- *--secret-key* (or *-k*): path of the symmetrical private key that is used to encrypt the value of any parameter
- *--csv-output* (or *-o*) <file.csv>: generates a CSV output of the execution, containing the objects successfully injected (see [page 18](#), « [Result of the execution](#) » chapter)

- `--csv-separator` (or `-P`) `<char>`: separator for CSV files. The default separator is the colon (':'). A single character is expected. This separator is used for both CSV file for errors and CSV output of the execution
- `--log-file` (or `-l`): name of the log file. If set, this option overloads the value of `log.filename` parameter set in `arcdefinjector.conf` configuration file
- `--verbose` (or `-V`): enables verbose mode
- `--version` (or `-v`): displays version
- `--help` (or `-h`): displays help

An example of command line is:

```
./launch.sh --arcsys-settings arcsys-settings.xml --archives sip.xml --csv-errors errors.csv
```

## 4.2. Result of the execution

The injector performs the following tasks:

- Checks the validity of the settings and the SIP file.
- For each `archive`, it creates: an archive, the permissions, the metadata, the objects, and the archiving request associated with the specified agent and engine. This request is "virtual" and is not processed by any component of Arcsys. It has only two statuses: `WORKING` and `ENDED` (or `ERROR` in case of an error).
- Optionally generates a CSV file containing errors if the `--csv-errors` parameter of the command line is defined. The structure of this CSV is as follows: `BASCODE` (the repository code), `COLCODE` (the collection code), `LOTCODE` (the lot code), `ERROR` (the technical code of the error, which can be sent to the Technical Support if needed), `MESSAGE` (the error message to process).
- Optionally generates a CSV file containing the list of lots and objects injected with success, if `--csv-output` parameter of the command line is defined. The structure of this CSV is as follows: `BASCODE` (the repository code), `COLCODE` (the collection code), `REQID` (the request identifier), `LOTCODE` (the lot code), `LOTID` (the lot identifier), `MOVID` (the sequence identifier), `OBJID` (the object identifier), `OBJNAME` (the object name), `SIZE` (the object size), `HASHFUNC` (the hash function of the object), `HASH` (the hash value of the object).
- Generates a log file containing the identifiers of the generated archives, requests, objects, the names of metadata (but not the values, in compliance with GDPR rules).
- Exits with the following return codes: 0 if all injections are successful, 1 if errors are encountered, 2 if errors are encountered and the `injection.errors.max` limit is reached.

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

### 4.3. Reentrance of the execution

The injector is reentrant. When iterating over archives defined in the SIP, if a lot code already exists in the database, it is skipped, and it is not counted as an error.

### 4.4. When the injection is done

It is strongly recommended to perform check requests on the archives of the targeted filesystem because the checksum of the files is not verified during the injection.

### 4.5. Identifying archives by reference

Archives by reference are identified by a field in the requests table, for future use (REQREF field).

# Glossary

## Access Zone

An access zone is an independent entity within Arcsys that defines a controlled network area from which resources can be accessed. These entities can then be attached to permissions (at the repository, collection, lot, or class level) to restrict or grant access based on the client's IP address when authenticating to the Arcsys REST API, the Arcsys Web Agent or ArcWeb Module.

## API (*Application Programming Interface*)

The APIs provided by Arcsys enable the product holder to fully customize a new application or user interface according to the specific ergonomic needs of their use case. Arcsys exposes several types of APIs:

- REST APIs are the recommended interface. They offer broad coverage of Arcsys's functionalities, including administration, operations, archiving, search, and archive retrieval.
- Legacy APIs based on RMI and SOAP protocols are still available for compatibility purposes but are deprecated and should no longer be used in new developments.

## 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.

## Archiving By Reference

Archiving by reference is a method in which data remains in its original storage location when added to an archive system, and the system generates references and metadata entries for the files. Eventually, the files are transferred to the archive system's defined storage using the copy and migration mechanism.

## 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](#).

## 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

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

complete (misleadingly called a "complete retrieval") or partial (*Partial Archive Retrieval*, misleadingly called a "partial retrieval").

See Also **Archive Restitution**.

## 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, which are: Arcsys Database, the Arcsys RMI, TCP/IP and SOAP API, the Arcsys REST API, the Arcsys Transfer Server, the Arcsys Transfer Service, the Arcsys Engine, the Arcsys Web Agent, the Arcsys Application Agent, the Arcsys Auto-Archive Agent, the ArcFF format control module, the CopyRequestManager, the Arcsys standard Clients, the ArcsysFsComparator File systems comparator, the ArcProofFolder Proof Folder module and the ArcsysBatchs batch module. See Also **Arcsys**.

## Arcsys Engine

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)
- ArcMover Tape Option (media manager managing file systems and tape libraries)
- 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)

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

- ArcCrypt Option (encryption of data at rest)
- ArcCFN (digital vault)
- ArcREF Option (record ingestion by reference)
- ArcMOVS3 Option (media manager allowing to archive and retrieve data on any Cloud media compatible with the Amazon S3 REST API)

### 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 in archiving and digital preservation refers to an organized framework for categorizing records and archival materials based on a hierarchical structure. It facilitates systematic retrieval, management, and preservation of information. In the context of Arcsys, the classification scheme is defined as the structural entity that contains a hierarchy of classes. These classes are used for organizing archives and records and for implementing specific archival policies such as retention schedules and format management. Within Arcsys, a classification scheme is linked to a specific repository, providing an organizational backbone for multiple collections. It also serves as a navigational tool for end users, enabling them to explore archives through the hierarchical structure of classes, alongside navigation by repository and collection.

### 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**.

### 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

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

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

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

### **Disposal due date** (or retention end date)

Scheduled end of retention date.

### **Disposal Hold**

Arcsys can be used to place a "disposal hold" on one or more lots archived in the application. This prevents certain sensitive operations, such as transitioning the lots to end-of-life status or migrating them to a different storage medium. All other operations remain authorized. The disposal hold guarantees that no irreversible change affecting the archival integrity or status of the lot can occur while the hold is active.

### **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 information of where the envelope is physically stored 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**.

### **Feature preview**

A Preview status on a feature enables early access to non-production features, allowing users to explore and provide feedback for improvement.

Features in Preview status should not be used in production environment, as they are not fully implemented yet.

### **Fixity**

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

### **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, the hash value 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

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

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**

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.

### **Journal**

A journal is an XML file which contains a list of PREMIS events.

### **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.

### **Lot enrichment**

The process of adding new objects to an existing archive.

### **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. The manifest 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.

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

### **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**.

### **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.

### **Permissions**

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

### **Program exit**

Place in the standard workflow for picking up and executing specific code.

See Also **Workflow**.

### **Proof folder**

A proof folder consists of a record, a proof slip, and, where appropriate, additional items (common signature or timestamp response, for example) that are used, by demonstrating the fixity and the authenticity of a document, for admission as proof. A proof slip can be generated using Arcsys Web Agent, ArcWeb Module, or Arcsys REST API. A proof folder can only be generated using ArcEP.

### **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** (or 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**

Logical entity in the Arcsys relational 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.

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

### **Restore**(or 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. The retention period 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. The retention start date 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 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 pool**

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

### **Storage zone**

The storage zone is a logical entity representing a physical storage space (e.g. set of file systems, tape libraries, cloud storage).

### **Synchronous retrieval**

Archive retrieval that takes place in the form of a direct retrieval of a document (for direct viewing or downloading) in a Web browser.

	<b>Arcsys</b>	ARCCO- EN12-25.3.2.STS-0
	ArcREF Option Guide	

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**

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.

**Transit Zone**

Entity logically connecting an application agent and a directory, along with additional configuration.

**Workflow**

A set of operations carried out from the beginning to the end of a process. In Arcsys, this refers to all actions carried out on archives and objects, either directly or indirectly, in the case of archives, from their pre-archiving or preparation to their removal from the system (after they have reached end-of-life). There are other workflows in Arcsys besides the archiving workflow, which are more administration-oriented. Customized workflow involves the use of at least one drop-off point to carry out customer processing.

## 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>

[infotel.com](https://infotel.com)