

EMBEDDED ARCHIVE STORAGE MIGRATION GUIDE

DOCU VERSION 1.3

MAY 5, 2014

otris software AG
Königswall 21
D-44137 Dortmund

www.otris.de

© Copyright 2014 otrs software AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means without express written permission of otrs software AG. Any information contained in this publication is subject to change without notice.

All product names and logos contained in this publication are the property of their respective manufacturers.

EASY reserves the right to make changes to this software. The information contained in this manual in no way obligates the vendor.

Table of Contents

1.	General Information.....	4
2.	Performing Migration.....	5
2.1	Basic assumptions	5
2.2	Performing migration	5
2.2.1	Minimum procedure	5
2.2.2	Naming fields.....	6
2.2.3	Migration from EE.i configuration	7
2.2.4	Parallel processing.....	7
2.2.5	Resuming migration.....	8
2.3	Notes on runtime behavior	8
2.3.1	Determining factors.....	8
2.3.2	Tweaks	9
3.	Call Parameters	10
3.1	Add source information (add-source)	10
3.2	Name of source archive (archive).....	10
3.3	Configuration file (config file).....	10
3.4	Debug mode (debug).....	10
3.5	Character set (encoding)	10
3.6	Exclusive index access (exclusive-index-access).....	10
3.7	Help texts (help)	10
3.8	Container files (import)	11
3.9	Batch size for indexing (index-batch-size)	11
3.10	Field mapping (mapping).....	11
3.11	Hit list for field mapping (mapping-ini-list).....	11
3.12	Attachment indexing (no-attachments)	12
3.13	Duplicate checking (record-check)	12
3.14	Batch size for registration (registry-batch-size).....	12
3.15	EE.i ini file (source-ini)	12
3.16	Spot check (spot-test).....	12
3.17	Name of location (storage-location).....	12
3.18	Threads (threads)	13
3.19	Status message (verbose).....	13
4.	Appendix: New Function and Parameters	14
4.1	Version 1.0.12.....	14

1. General Information

This document describes how to migrate container files of an EASY ENTERPRISE.i archive (EE.i archive) to a store of the **Embedded Archive Storage** (also called EDA) using the migration tool *ecftoeas.exe*.

It initially describes ideal-typical procedure while migrating EE.i containers, as well as existing experience and tweaks. This is followed by a description of the call parameters.

2. Performing Migration

2.1 Basic assumptions

The development of the migration tool was based on some basic assumptions that have an influence on how to perform migration of EE.i archives. These basic assumptions are:

1. Only the container files of an EE.i archive are migrated.
2. There is no automatism where names for file types are inferred from a location or by naming container files.

2.2 Performing migration

2.2.1 Minimum procedure

ECF container files are always migrated to an existing EAS store, i.e. new stores are not created by the migration tool `ecftoeas.exe`. The default administration tool `eas.exe` or the Create-Store entry on the Windows start menu can be used to create new stores (see also EAS system documentation):

```
eas.exe -c \path\to\store1.ini -i
```

To migrate the content of an ECF container file to this created store, you can call the migration tool as follows:

```
ecftoeas.exe -c \path\to\store1.ini -i FREEDB.ECF -r file type
```

This call ensures that the entries that the container file `FREEDB.ECF` contains migrate to Store1, where these files are assigned the "File type" file type during migration. The `-c` parameter specifies the path to the ini file to which the container file is to migrate; the `-i` parameter specifies the path of the container file to be migrated, and the `-r` parameter specifies the file type to be assigned.

These are minimum specifications required for migrating a container file, so you need to specify all three parameters. In this manner, you can migrate individual container files, target-oriented, to the respectively intended EAS stores.

In addition, initial variations of the call are available, so you do not need a separate call for each individual container file:

```
ecftoeas.exe -c \path\to\store1.ini -i FREEDB.ECF FRDB0001.ECF -r file type
```

Here the two container files `FREEDB.ECF` and `FRDB0001.ECF` are migrated to EAS store Store1, where they are processed in the sort order.

Important note on DOCUMENTS file versions!

To obtain the version history of DOCUMENTS files contained in the container files, you need to consider the order of migrating container files, i.e. you need to start migration with the first container file.

The migration tool therefore supports adherence to the order in that container files are sorted by name prior to migration and processed in the sorted order during migration.

If you want to migrate all container files through a single call, you can also use placeholders:

```
ecftoeas.exe -c \path\to\store1.ini -i FR*.ECF -r file type
```

Here all container files whose name starts with "FR" are migrated to the EAS store Store1.

Important note on the file type!

All DOCUMENTS files created in EAS from the container files are assigned the file type specified by -r. If the container files contain DOCUMENTS files of a separate type, you will have to call ecftoeas.exe separately for these containers.

2.2.2 Naming fields

The file fields in the container files do not have any meaningful names, but only numbers, e.g. 1001, 1002, etc. These are directly transferred to the EAS store.

However, fields in EAS may have meaningful names, which may be necessary. If you want to convert the field names to meaningful names during migration, you will have to specify a mapping file when calling them, using the --mapping or -m parameter:

```
ecftoeas.exe -c \path\to\store1.ini -i FR*.ECF -r file type ↵  
-m \path\to\mapping.fld
```

The mapping file is a simple text file where each line defines how a field is renamed from the container file. For this purpose, each line provides, separated by tabulators, the field number from the container file, the target name and target file type, as well as the optional analyzer.

Number	Name	Analyzer
--------	------	----------

When indexing the fields, different conversions are made which facilitate quick search in different variations. In some cases, however, this may be undesirable, e.g. with identification numbers such as invoice numbers. In that case, you can specify an analyzer that specifies the conversion, using the third value. Currently, only one analyzer is available, i.e. the `keyword` analyzer. This indicates a field as an identification field, ensuring that the value of this field is indexed unchanged.

Additionally, the data itself is also converted in specific cases. This particularly affects date values and numeric values. This is automatically performed; however, it can also be explicitly defined by adding the data type to the number. Separation is via a caret character and specifying the data type. The structure of the file is then as follows:

Number^Type	Name	Analyzer
-------------	------	----------

Three data types are available:

- **date:** Field values are stored as date values in the EE.i archive.
- **number:** Field values are stored as numeric values in the EE.i archive.
- **text:** Field values are stored as textual values in the EE.i archive.

Moreover, a specific data type is available:

- **ignore:** Field is not transferred during migration.

A sample mapping could then look as follows:

1001	Number	keyword
1002	Artist	
1003	Album	
1004	Year	
1005	Genre	
1006	Content	
1007	CreationDate^date	

2.2.3 Migration from EE.i configuration

Alternately, some of the necessary data from the EE.i archive's original ini file can be read by the migration tool. To do this, however, you need to specify the path to the ini file of the EE.i archives. The migration tool then reads both the containers to be migrated and the field mapping from the EE.i archive's configuration.

A call using this parameter would look like this:

```
ecftoeas.exe -c \path\to\store1.ini -r file type ↵
--source-ini \path\to\EE.i\FREEDB.ini
```

This call ensures that all ECF files of the EE.i archive FREEDB are migrated and the field mapping defined in the STANDARD hit list is used.

If you want to use another hit list from the EE.i archive's ini file, then this can be specified through the additional parameter `--mapping-ini-list`. This call would then look like this:

```
ecftoeas.exe -c \path\to\store1.ini -r file type ↵
--source-ini \path\to\EE.i\FREEDB.ini --mapping-ini-list LIST
```

2.2.4 Parallel processing

In the above examples, the container files are migrated in order. However, this can also be processed in parallel. This can be enabled via the `--threads` parameter using the short form `-t`, where you can specify the number of threads to perform migration:

```
ecftoeas.exe -c \path\to\store1.ini -i FR*.ECF -r FileType -t 2
```

In this call two threads process the container files. These are spread evenly over both threads, where the version history is considered. Should different versions spread over multiple container files and therefore over multiple threads, the threads will automatically synchronize, so the version history is preserved.

2.2.5 Resuming migration

If you have canceled migration in the meantime, you can resume it without migrating EE.i files that have already been migrated a second time. To enable the validation necessary for this, you need to additionally specify the `--record-check` parameter for the call:

```
ecftoeas.exe -c \path\to\store1.ini -i FR*.ECF -r file type --record-check
```

By default, this function is not active because it involves the corresponding performance losses.

2.3 Notes on runtime behavior

2.3.1 Determining factors

Depending on the data volume of an EE.i archive, we cannot make a definite statement on how long migration will take. Its duration is in fact influenced by the structure and the type of content of the containers.

- Proportionally brief runtimes are obtained when migrating only few EE.i files containing few but very large attachments within the containers. In a test case, container files comprising 3.6 GB and containing about 9,000 EE.i files and 70,000 attachments were migrated to an empty store in 10-15 minutes using default parameters.
- Whereas proportionally long runtimes are obtained when the containers contain only EE.i files without attachments. In a test case, container files comprising about 2.4 GB and about 2 million EE.i files were migrated to an empty store in 3- 4 hours using some tweaks.

The tests were performed on a computer with a 3.3 GHz six-core processor, 16 GB RAM and SSD disk.

2.3.2 Tweaks

Migration can speed up via different settings. However, the price for these is loss of security because total crashes, e.g. of the complete computer, require to restart migration all over again or perform it with existence check enabled.

- Parallel processing (`threads`): As described above, containers do not need to migrate sequentially, they can also migrate in parallel. Generally, more threads result in shorter runtimes. Because a container file is always assigned to a thread, the maximum number of threads should therefore match the number of container files to migrate.
- Exclusive index access (`exclusive-index-access`): Ensures that the index is open during the entire runtime. This eliminates the need for opening and closing the index during file indexing; however, it locks out any other thread to write in the index during that time. Normal archiving is not possible during that time and in cases of total crashes the index may be corrupted. See also chapter 3.5.
- Index batches (`index-batch-size`): Ensures that entries for the index are gathered in RAM first, and then written in a block. This eliminates the need for pre- and post-processing steps which you would otherwise have to perform for each EE.i file or attachment individually. In case of a total crash, this may require migration to be started all over again.
- Registry batches (`registry-batch-size`): Analogous to the index batches, only for the registry. Essentially, the same applies here.
- Disabling attachment indexing (`no-attachments`): Suppresses indexing attachments during migration. This can then be subsequently performed through normal index restoration. Alternately, `delayedRecordIndexing` and `delayedAttachmentIndexing` can also be used to suppress indexing and subsequently perform it.

3. Call Parameters

3.1 Add source information (add-source)

This parameter is used to specify that additional information about the source archive is added to the migrated EE.i files. This includes the name of the source archive and the location of the EE.i file. The name is inserted into the file field `otrisSourceArchive` and the location into the `otrisSourceStorageLocation` field.

This data is determined either from the ini file of the EE.i archive if it has been specified, or from the values specified with the `--archive` and `--storage-location` parameters.

```
--add-source
```

3.2 Name of source archive (archive)

This parameter is used to specify the name of the source archive which is also to be stored in the migrated EE.i files as additional information.

```
--add-source --archive ARCHIVE
```

3.3 Configuration file (config file)

This parameter is used to specify the path to the configuration file to which to migrate an EE.i archive. Specifying this parameter is necessary.

```
--config-file \path\to\store.ini  
-c \path\to\store.ini
```

3.4 Debug mode (debug)

Enables debug mode, which produces more outputs that may be of use for error search. Specifying this parameter is optional for migration.

```
--debug
```

3.5 Character set (encoding)

Allows specifying the character set in which the texts are encoded within the EE.i container. If the character set is not specified, the currently set character set of the operating system will be used.

```
--encoding=UTF-8
```

3.6 Exclusive index access (exclusive-index-access)

Enables exclusive index access during migration, i.e. the migration tool exclusively locks the index during the complete migration runtime. This increases speed because opening and closing the index is obsolete; however, no other process can write in the index and, in case of a complete computer crash, the created index can be more easily corrupted.

```
--exclusive-index-access  
-x
```

3.7 Help texts (help)

Outputs all call parameters with brief explanations.

```
--help  
-h
```

3.8 Container files (import)

The container files to be migrated are specified through this parameter. The files can be explicitly named, or implicitly using wildcards. Specifying this parameter is mandatory.

```
--import \path\to\container.ecf  
-i \path\to\container.ecf  
-i \path\to\*.ecf  
-i \path\to\container1.ecf \path\to\container2.ecf
```

3.9 Batch size for indexing (index-batch-size)

Sets the size of the packages to be processed at once during indexing. By default, each DOCUMENTS file and attachment are immediately indexed. If, however, a value is entered here, these DOCUMENTS files and attachments will be gathered and only indexed when the number has reached the specified value, and will be concurrently written in the index.

This may cause runtime to be reduced because everything is initially written in memory, and then in the block in the index. This makes many pre- and post-processing steps which would otherwise have to be performed for each individual EE.i file or attachment unnecessary.

Specifying this parameter is optional for migration.

```
--index-batch-size <Number>
```

3.10 Field mapping (mapping)

Specifies the name of the file with the field mapping. Specifying this parameter is optional for migration.

```
--mapping \path\to\mapping.fld  
-m \path\to\mapping.fld
```

If at the same time you specify a mapping with `--mapping-ini-list`, then the field mapping specified with `--mapping` is valid.

3.11 Hit list for field mapping (mapping-ini-list)

Specifies the name of the hit list to be used for field mapping. This parameter is only effective in combination with the `--source-ini` parameter. The hit list to be specified must be defined in the EE.i archive's ini file.

Specifying this parameter is optional. If you do not specify it, the STANDARD hit list from the EE.i archive's specified ini file will be used.

```
--source-ini \path\to\EE.i-archive.ini --mapping-ini-list LIST
```

If at the same time you specify a mapping with `--mapping`, that mapping is valid.

3.12 Attachment indexing (no-attachments)

Disables indexing attachments during migration. This can speed up the migration process; however, subsequent indexing will be necessary after migration is complete if you want to be able also to search in attachments.

Specifying this parameter is optional for migration.

```
--no-attachments
```

3.13 Duplicate checking (record-check)

Enables duplicate checking during migration. In that case, a check is made for each EE.i file on whether it has already been migrated to the destination store. Specifying this parameter is optional for migration.

```
--record-check
```

3.14 Batch size for registration (registry-batch-size)

Analogous to the batch size for indexing, this parameter sets the size of the packages to be written together in the registry.

This may cause reduction of runtime because everything is written in memory first, and then in block in the registry. This makes many pre- and post-processing steps which would otherwise have to be performed for each individual EE.i file or attachment unnecessary.

Specifying this parameter is optional for migration.

```
--registry-batch-size <Number>
```

3.15 EE.i ini file (source-ini)

This parameter allows specifying the path to an ini file of an EE.i archive to be migrated. If this is specified, the container file will on the one hand be read from the ini file, and the field mapping will be used from this ini file on the other.

```
--source-ini \path\to\EE.i-archive.ini
```

3.16 Spot check (spot-test)

Simulates the migration process and determines spot checks (10) of field values for each field. If you specify a file, the spot checks will be written in that file. This is intended for such cases where the actual naming of fields in the container files is no longer known. In that case, random data is output from these fields, so you can derive possible meaningful names from the content.

Specifying this parameter is optional for migration.

```
--spot-test \path\to\report.txt
```

3.17 Name of location (storage-location)

This parameter is used to specify the name of the location to be stored as an additional piece of information with the migrated DOCUMENTS files.

```
--add-source -- store-location LOCATION
```

3.18 Threads (threads)

Specifies the number of threads that migrate the containers in parallel. Specifying this parameter is optional for migration.

```
--threads <Number>  
-t <Number>
```

3.19 Status message (verbose)

Enables outputting status messages during migration. In doing so, you need to specify a level. The higher that level is the more outputs are created. Specifying this parameter is optional for migration.

```
--verbose <Number>  
-v <Number>
```

4. Appendix: New Function and Parameters

4.1 Version 1.0.12

New call parameters:

- `--add-source` (described in chapter 3.1)
- `--archive` (described in chapter 3.2)
- `--spot-test` (described in chapter 3.16)
- `--storage-location` (described in chapter 3.17)