

# FI-AA - Asset Accounting: Data Transfer Workbench



HELP.FIAA

**Release 4.6B**



## Copyright

© Copyright 2000 SAP AG. All rights reserved.

No part of this brochure may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.

ORACLE® is a registered trademark of ORACLE Corporation, California, USA.

INFORMIX®-OnLine for SAP and Informix® Dynamic Server™ are registered trademarks of Informix Software Incorporated.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of The Open Group.







HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Laboratory for Computer Science NE43-358, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139.

JAVA® is a registered trademark of Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94303 USA.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, mySAP.com, mySAP.com Marketplace, mySAP.com Workplace, mySAP.com Business Scenarios, mySAP.com Application Hosting, WebFlow, R/2, R/3, RIVA, ABAP, SAP Business Workflow, SAP EarlyWatch, SAP ArchiveLink, BAPI, SAPPHIRE, Management Cockpit, SEM, are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

## Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

## Contents

<b>FI-AA - Asset Accounting: Data Transfer Workbench .....</b>	<b>5</b>
<b>Process for Automatic Asset Data Transfer .....</b>	<b>7</b>
<b>Structuring Rules for Transfer Data .....</b>	<b>10</b>
<b>Description of Transfer Format BALTD (Master Data and Values) .....</b>	<b>11</b>
Modification of Transfer Format BALTD.....	14
<b>Description of Transfer Format BALTB (Transactions).....</b>	<b>15</b>
Modification of Transfer Format BALTB.....	17
<b>Field Types of the Transfer Format .....</b>	<b>18</b>
<b>Rules for Setting Up the Input File.....</b>	<b>19</b>
<b>Batch Input and Direct Input.....</b>	<b>23</b>
<b>Legacy Data Transfer Using Batch Input .....</b>	<b>25</b>
<b>Error Correction for Batch Input .....</b>	<b>26</b>
<b>Direct Input.....</b>	<b>28</b>
<b>Error Correction for Direct Input.....</b>	<b>31</b>

# FI-AA - Asset Accounting: Data Transfer Workbench

## Definition

Automatic transfer of asset data from a legacy system to the R/3 System.

## Use

### Technical

There are two methods for automatic transfer of legacy asset data:

- Transfer of legacy data using a batch input procedure (program RAALTD01)  
When you use the batch input procedure, the system creates a batch input session containing the legacy asset data and automatically processes this session. The batch input session provides values to the transaction for asset data transfer in the FI-AA component. The system writes any transfer records, which have errors, to another batch input session. You have to process this second session manually. You enter a name for this error batch input session when you create the batch input session for the legacy data transfer.
- Transfer of legacy data using direct data import (program RAALTD11)  
The direct data input writes legacy data directly to the respective FI-AA tables. The system writes any errors that occur to an error file. You enter the name of the error file when you start the transfer. Direct data input offers the advantage of better performance compared to the batch input procedure. However, it does have limitations as regards the input data and protection against errors. For more information, see the R/3 library FI-AA "Data Transfer -> Direct Data Import."



When you have fewer than 10,000 assets, you should use the batch input procedure, since it can better ensure that data records with errors are not transferred into the R/3 System. Depending on your hardware, the batch input interface can normally transfer between 10 and 40 asset records (including their transactions) per minute into the FI-AA component.

For larger numbers of assets, we recommend the direct data input procedure. As compared to batch input, it offers a ten-fold increase in performance.

## Object-Specific Settings and Options

By setting the activity category, you determine whether the legacy data transfer is carried out using batch input or direct data input.

## Constraints

- The Data Transfer Workbench limits the transfer structure to a maximum of 5000 characters. Therefore, you can only use the Data Transfer Workbench when you have fewer than ten depreciation areas. If you have more than ten depreciation areas, the transfer structure BALTD (asset master data and values) might be longer than 5000 characters. In that case, you have to carry out the transfer using report RAALTD01 (batch input) or report RAALTD11 (direct data import) using transaction SE38.

---

**FI-AA - Asset Accounting: Data Transfer Workbench**

You can find out how long your BALTD transaction structure is using transaction SE11 (*Dictionary object* → *Display*). The header data and fields of the transfer structure are displayed on a screen you reach by choosing *Extras* → *Runtime object* → *Display*. Under the header data in column "Len" you see the length of your structure in number of characters.

- You cannot transfer net book values. You have to transfer the gross values for APC and accumulated depreciation.



Make sure that the R/3 maintenance level of the system you used to create the FI-AA data structure is the same as the R/3 maintenance level to which you want to transfer the data.

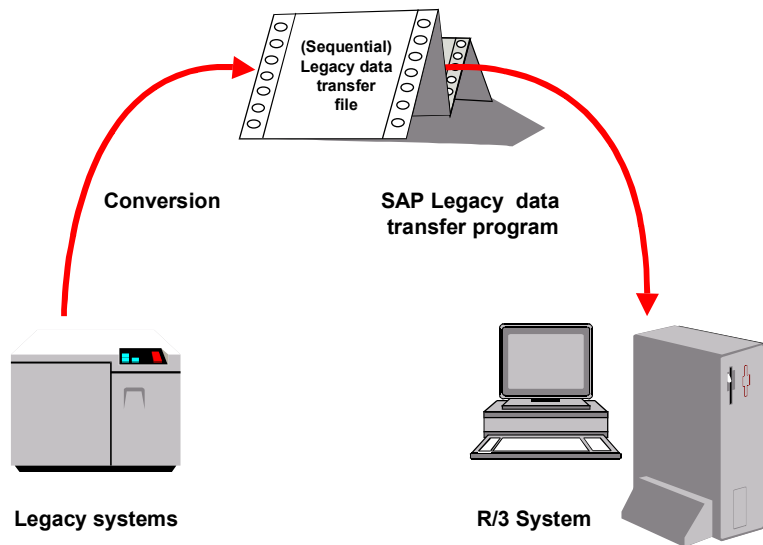
## Process for Automatic Asset Data Transfer

### Purpose

The following process flow can be used for transferring fixed assets from a legacy system to the R/3 System.

### Process Flow

To transfer legacy assets (master data and values or transactions), you convert the legacy data into a sequential transfer file, and then transfer it to the R/3 System, either using report RAALTD01 (batch input) or RAALTD11 (direct data import). The transfer file contains the data in the format that is required by SAP for a successful data transfer.



### Preparations

The following settings have to be made in Customizing for *Asset Accounting* in preparation for the legacy data transfer (choose *Asset Data Transfer*):

- Decide on the transfer date and the transfer options for each company code and/or depreciation area.
- Specify the sequence in which the depreciation areas are to be supplied with values. It makes sense to arrange the depreciation areas so that those areas with values that are not, or only partially, calculated by the system, appear first on the screen in dialog transactions.
- If you are using a year-dependent fiscal year version, you also have to carry out the following steps. For all historical fiscal years, from the capitalization year of your oldest asset minus 1, up to the current fiscal year, you have to
  - Maintain the calendar date assignments for the fiscal year variant (in Customizing for *Financial Accounting*).
  - Generate the depreciation period controls for Asset Accounting (in Customizing for *Asset Accounting*, choose *Periodic processing*).

### Execution

---

**FI-AA - Asset Accounting: Data Transfer Workbench**1. [Identify relevant fields \[Ext.\]](#)

You can display the asset master record using transaction AS91 (create legacy asset). Use an asset class with a screen layout that displays all fields.

2. [Analyze transfer structure \[Ext.\]](#)

Modify the transfer structures BALTD (master data and values) and BALTB (transactions) that are supplied by SAP, if modification is necessary. You need to make modifications if you want to transfer more than eight depreciation areas, since the BALTD structure provided only allows for eight depreciation areas. The same applies to transactions and their posting amounts in the depreciation area.

3. [Example- Create sample data transfer file \[Ext.\]](#)4. [Test transfer program \[Ext.\]](#)5. [Analyze legacy data \[Ext.\]](#)6. [Assign fields \[Ext.\]](#)7. [Write conversion program \[Ext.\]](#)

You might want to generate a record structure from the R/3 transfer structures in a particular programming language. To do so, you have to make the following entries:

- Enter the transaction.
- Enter **BALTD** (master data and values) or **BALTB** (transactions) in the *Table name* field.
- Enter **AM\_ANLA** in the *Identifier in TSRCG* field.
- Enter the full name of the file to which the system should transfer the data structure.

8. [Prepare legacy system \[Ext.\]](#)9. [Test transfer \[Ext.\]](#)

Before an update run or test run of the transfer program, you **must** carry out a preliminary sort of the transfer records in the transfer file according to their keys.

After a test transfer of fixed assets, you can reset the company code (in Customizing for Asset Accounting, choose *Preparing for Production Startup* → *Production Startup* → *Reset Company Code*).

10. [Carry out transfer \[Ext.\]](#)**Postprocessing**

- Complete the transfer by comparing the values of your legacy system with the values in SAP R/3 Asset Accounting (reconciliation of balances).
- Then block the legacy asset data transfer transactions. You block them at the company code level, since you might be transferring asset data for different company codes at different points in time.

The following status settings are possible in the Customizing definition of the company code:

- The company code is in test operation (transfer of legacy asset data is allowed at any time, also parallel to transactions).

FI-AA - Asset Accounting: Data Transfer Workbench

- Transfer of legacy asset data is not yet finished (asset transactions are not possible).
- Transfer of legacy asset data is finished (asset transactions are possible).

---

**Structuring Rules for Transfer Data**

## Structuring Rules for Transfer Data

Master data that is to be transferred automatically must be available in a sequential input file and be structured according to the following Data Dictionary tables:

- Master data/values: Table BALTD
- Transactions: Table BALTW

## Description of Transfer Format BALTD (Master Data and Values)

The master data to be transferred has to be in a sequential input file and be structured according to Dictionary structure **BALTD**:

*EOR* | Key | Master data | Area-01 | ... | Area-08 | *EOR*  
(*EOR* = end of record)

You can see the exact structure of the transfer record by choosing *Goto* → *Analyze Files and Data Structures* → *Structure* → *Display* in the Data Transfer Workbench. You can create a record layout for this structure with the programming languages COBOL, C, ASSEMBLER and PL1 in the Data Transfer Workbench.



Make sure that the R/3 release with which you create the data structure is the same as the release to which the data is transferred.

The individual record parts have the following meanings:

- **Key**

The key consists of these fields:

- MANDT (client)
- BUKRS (company code)
- ANLKL (asset class)
- OLDN1 (legacy asset main number)
- OLDN2 (legacy asset sub-number)
- TCODE (accessed transaction)
- RCTYP (record type)

For master data you should always enter record type **A** In the **TCODE** field, you have to specify the transaction for which batch input should be created:

- AS91 (create legacy asset)
- AS92 (change legacy asset)
- AS94 (create sub-number for legacy asset)
- AS81 (create legacy group asset)
- AS82 (change legacy group asset)
- AS84 (create sub-number for legacy group asset)

- **Master data**

The fields of the asset master record which are independent of the depreciation areas are in the general asset master data. In particular you should specify the main number and the sub-number of the new asset master record here, if you provided for external number assignment in the number range or in the asset class.

## Structuring Rules for Transfer Data

Enter the number of transactions that should be transferred with the asset in the field BWCNT (refer to [Description of the Transfer Format BALTB \(Transactions\) \[Page 15\]](#)).

Transfer investment keys into the fields INVSL01, INVSL02,... and so on.

The cost center of an asset cannot be transferred with time dependence. It is only possible to transfer the cost center that is currently valid. You can enter a cost center history after the transfer, using the legacy data transaction (AS92).

- **Area-01 to area-08**

The fields of the block for area-*nn* can be recognized by the suffix *nn* in the field name. They contain the entries to be transferred for the depreciation area into the field AFABEnn, that is

- Depreciation terms and
- Transferred values

The sequence of the fields is identical in every area-*nn*.

### Depreciation Terms

AFABG01 ordinary depreciation start date

ZINBG01 start date for interest calculation

SAFBG01 special depreciation start date

AFASL01 depreciation key

NDJAR01 useful life (years)

NDPER01 useful life (periods)

WBIND01 index series

ALIND01 index series (age-dependent)

APROP01 variable depreciation portion

UMJAR01 changeover year

NDURJ01 original useful life (years)

NDURP01 original useful life (periods)

SCHRW01 scrap value

NDABJ01 expired useful life (years)

NDABP01 expired useful life (periods)

ANDSJ01 expired useful life of special depreciation (years)

ANDSJ01 expired useful life of special depreciation (periods)

ANLGR01 group asset (main number)

ANLGR201 group asset (sub-number)

INBDA01 date ready for start-up

VYEAR01 acquisition year

VMNTH01 acquisition month

Structuring Rules for Transfer Data

RSTCK01 remaining units depreciated

**Values**

KANSW01 cumulative acquisition value

KAUFW01 cumulative revaluation of replacement value

KINVZ01 cumulative investment support

KNAFA01 cumulative ordinary depreciation

KSAFA01 cumulative special depreciation

KAFA01 cumulative unplanned depreciation

KMAFA01 cumulative transfer of reserves

KAUFN01 cumulative revaluation of ordinary depreciation

KZINW01 cumulative interest

AUFWB01 posted revaluation of replacement value

NAFAG01 posted ordinary depreciation

SAFAG01 posted special depreciation

AAFAG01 posted unplanned depreciation

MAFAG01 posted transferred reserves

ZINSG01 posted interest

AUFNG01 posted revaluation of cumulative ordinary depreciation

## Structuring Rules for Transfer Data

# Modification of Transfer Format BALTD

Transfer format **BALTD** is provided as a proposal which can be modified.

- **Modification of Master Data Block**

Here you can delete unnecessary fields or add additional fields. When adding fields, however, make sure that a corresponding screen field exists in the transfer transaction for the field being added. (If necessary, modify a screen.)

The only exceptions are fields INVSL01 and INVSL02. You are **not** allowed to remove them from the transfer structure.

- **Additional INVSLnn Fields**

In the standard system, two fields, INVSL01 and INVSL02, are intended for the transfer of investment keys. However, up to 99 fields are possible. If you need more fields, you simply need to add them to the Dictionary table **BALTD**. Add these fields after the INVSL02 field, and name them INVSL03, INVSL04,... and so on.

- **Increase/Reduction of the Area-*nn* Blocks**

The number of the 8 blocks, area-01 to area-08, is a default. You can use fewer than 8 blocks, or add a maximum of up to 99 such blocks. When increasing the number of these blocks, you should bear in mind two rules.

- The setup of the blocks (fields and their sequence) must always be the same.
- The number sequence of the blocks, starting with 01, has to be continuous and unbroken (contiguous), and the fields of area *nn* all have to have the suffix *nn*.



You do **not** need to adapt transfer program **RAALTD01**.

## Description of Transfer Format BALTB (Transactions)

The transactions to be transferred have to be structured according to Dictionary structure

**BALTB:**

*EOR* | Key | Transaction data | Amount 01 | ... | Amount 08 | *EOR*

(*EOR* = end of record)

All transaction records for a fixed asset have to be structured in the same way as the transfer record for the master data of the asset (Dictionary structure **BALTD**). The transaction records are separate data records. In the input file, make sure they are separated from the previous record by an end of record indicator.

The structure **BALTD** for the transfer of master data and values (refer to [Description of the Transfer Format BALTD \(Master Data and Values \[Page 11\]\)](#)) contains the number of transaction records that belong to a master data transfer record. This information is contained in field **BWCNT**:

*EOR* **Master data record** *EOR* **Transaction record1** *EOR* **Transaction record2** *EOR* ... *EOR*

You can also use the Data Transfer Workbench to create a record structure in various programming languages for table **BALTB**.



Make sure that the R/3 release with which you create the data structure is the same as the release to which the data is transferred.

The individual record parts have the following meanings:

- **Key**

The key consists of these fields:

- MANDT (client)
- BUKRS (company code)
- ANLKL (asset class)
- OLDN1 (legacy asset main number)
- OLDN2 (legacy asset sub-number)
- TCODE (accessed transaction)
- RCTYP (record type)

For transactions always enter record type **B**.

- **Transaction data**

The general transaction data consists only of two fields:

- BWASL (transaction type)
- BZDAT (asset value date)

- **Amount 01 to amount 08**

Amounts belonging to the depreciation areas **AFABEnn** are transferred to the fields **ANBTRnn**. You can set up certain depreciation areas in *Customizing for Asset Accounting* so that they adopt their posting values from a reference depreciation area. You do not have to

### Structuring Rules for Transfer Data

enter values for such depreciation areas, unless they differ from the amounts in the reference depreciation area.



Depreciation area **01** is managed in the block for area **01**, and area **20** is managed in the block for area **03**, (this means. in the BALTD structure, AFABE01 = **01** and AFABE03 = **20**). If area **20** adopts its posting values from area **01**, then you only need to transfer values for the ANBTR03 field if you want the values in area **20** to be different from the values in area **01**.

## Modification of Transfer Format BALTB

You only need to modify transfer format **BALTB** if you changed the number of blocks for area-*nn* in transfer format **BALTD**. If this is the case, you have to increase or decrease the fields ANBTR*nn* accordingly (refer to [Modification of Transfer Format BALTD \[Page 14\]](#) ).

## Structuring Rules for Transfer Data

# Field Types of the Transfer Format

The fields of the transfer record are all of the type CHAR or DATS.

- **CHAR (Character)**

These fields are filled with the same contents as in online entry. All value fields of CHAR type have 16 places, so that you can transfer 10000 as 10000 or 10,000 or 10,000.00. In the same way, a useful life of 3 years can be transferred as 3,03 or 003.

When you use direct data import, the fields that use alpha conversion (such as cost center) have to be provided with leading zeros, if the strings being transferred are numerical only (refer to [Direct Data Import \[Page 28\]](#)).

- **DATS (Date)**

All date fields are type DATS. They must be completed according to the format `yyyymmdd`, that is, **not** according to the format for online entries. The system rejects entries that are in the normal online format.



If you process the input file using the editor in the Data Transfer Workbench, then dates appear in the normal online format (MM/DD/YYYY for USA). Enter changes in the same way.

## Rules for Setting Up the Input File

### Use

There are certain rules for creating the input file that *must* be observed. These rules are explained below.

### Features

#### Identification of Fields Not To Be Transferred

There may be fields that should not be transferred (for example, because they are to be supplied with a default value from the sample master record, or because they should remain empty). You must mark any such fields with the slash character "/" in the first position. If, for example, the depreciation key in area 01 is not to be transferred exactly, enter / in AFASL01.



Do not use the symbol "#" in the input file (not even in text fields). This symbol is interpreted as a control indicator by the system during processing of the batch input file.

#### Initial Record Layout

If the program for setting up the input file is an ABAP program, you can use an include to help initialize the fields for batch input. The include RAALTDI0 contains the form routine BALTD\_INIT. This form routine creates a record in BALTD format that contains a "/" in the first position of BALTD fields, in other words, a sort of "initial record layout" for batch input. For performance reasons, the form routine should not be called up for every record to be transferred using the transfer program. Instead, you should create this "initial record layout" once, and then keep it available in a help structure (in the example, it is the structure I-BALTD). Below is an example of how this could look:

#### Transfer program

...

TABLES: BALTD.

```
DATA: BEGIN OF I-BALTD.  
      INCLUDE STRUCTURE BALTD.  
DATA: END OF I-BALTD.
```

...

```
PERFORM BALTD_INIT USING I-BALTD.
```

...

```
INCLUDE RAALTDI0.
```

...

## Rules for Setting Up the Input File

You do *not* have to adapt the include RAALTDI0 if changes are made to table BALTD in the Dictionary.

### Handling of the Key

Enter the transaction, for which the batch input should be created, in the transaction code TCODE: AS91 (create old asset), AS92 (change old asset) or AS94 (create sub-number for old asset).

Always enter the desired asset class in the field ANLKL, *even* when working with the transactions AS92 and AS94.

When working with the transaction AS92, enter the asset class for the asset to be changed in ANLKL. If you are adding a sub-number using AS94, enter the asset class for the asset main number.

If the asset class is numeric, transfer it using leading zeros:

- asset class 1250 ==> BALTD-ANLKL = "00001250" but
- asset class AIB ==> BALTD-ANLKL = "AIB ".

In the fields OLDN1 and OLDN2, specify the main number and sub-number of the asset in your legacy system. As long as you are not transferring any transactions, these fields are not actually necessary for the creation of the new asset master record. However, it makes sense to specify them for two reasons:

- In the case of errors, the asset can be identified using the numbers from the old system. This is the only way to clearly identify the assets listed in the error log. The numbers from your legacy system are particularly important in case of a termination of the transfer run. The numbers enable you to determine the record with which the run should be restarted (refer to [The Transfer Run \(Test Possibilities\) \[Page 26\]](#)).
- If the new asset numbers are to be assigned internally, this takes place in the sequence determined by the old numbers, since the input file must be sorted according to the key before the running of the transfer program.

Specify the record type in the field RCTYP:

- RCTYP = **A** for master data (transfer structure BALTD),
- RCTYP = **B** for transactions (transfer structure BALTB).

### Special Concerns Regarding Transfer of Transactions

If the transfer date set in Customizing does not fall at the end of a fiscal year, you can also transfer acquisition transactions that have taken place since the beginning of the fiscal year. In this case you would transfer not only master data in the format BALTD, but also transaction data in the format BALTB.

It is possible to transfer transactions that were entered after the close of the last fiscal year and before the transfer date. However, you can only transfer acquisition transactions automatically. You have to enter retirements and transfer postings manually after the automatic transfer. You have to keep this in mind when entering the transaction type. You can recognize acquisition transaction types by the transaction type category 1 in the transaction type group. The following rules apply for the transfer of transactions:

- Record type: Always set the record type BALTD-RCTYP = **A** and BALTB-RCTYP = **B**.

### Rules for Setting Up the Input File

- Old asset numbers: In order to ensure the correct assignment of transactions to their assets, you *must* make sure that the fields BALTD-OLDN1, BALTD-OLDN2 are completed.
- Counter field for transactions: The field BALTD-BWCNT *must* always be completed and contain the exact (!) number of transferred transaction records for a legacy asset. For example, if two BALTB records were transferred for a particular asset, the field must be completed BALTD-BWCNT = **0002**. If there are no transactions at all transferred for a given asset, complete the field with **0000**.
- Transaction type: Since only acquisition transactions can be transferred, the field BALTB-BWASL can only be completed using acquisition transaction types.
- Asset value date: The asset value date BALTB-BZDAT must lie in the time period between the start of the fiscal year and the legacy data transfer date (as specified in FI-AA Customizing).
- When you transfer line items for assets under construction, do **not** enter value dates. The system automatically sets the value date to January 1 of the current fiscal year. You use the normal transaction types for acquisitions (such as 100) for line items that were acquired in the current fiscal year. For line items from previous fiscal years, use the special transaction types for acquisitions from previous years (900, 910). This method ensures that these line items are identified as acquisitions from previous years, despite the asset value date January 1 of the current year.
- No transactions can be transferred for transaction AS92.

### Sequence of Depreciation Areas For Asset Data Transfer

The following is only relevant if more than 4 depreciation areas are being transferred.

You can set up a special sequence, which applies only for the legacy asset data transfer, for the order of the depreciation areas in the cumulative value screen and transaction screens of the transactions AS91, AS92 and AS94. You set up the sequences per company code in FI-AA Customizing. A maximum of 4 depreciation areas can be entered in each of these screens in these transactions. Therefore, for performance reasons, you should arrange the sequence so that depreciation areas that do *not* take over (copy) values from other areas come first. Setting up the sequence in this way means that less scrolling is required in the data transfer transaction, since value takeover areas copy their values (either wholly or in part) directly from other areas.

### Transfer of Net Book Value

For technical reasons, it is **not** possible to transfer net book values using batch input transfer. If you nevertheless wish to transfer net book values from your legacy system, you have to set up the interface program in your legacy system with this in mind. Set up the interface program so that it already calculates accumulated depreciation as the difference between the acquisition value and the net book value. You can then transfer accumulated depreciation and acquisition costs to the SAP R/3 System.

### Handling of the Fields INVSLnn in the Transfer Structure BALTD for Master Data

If you use investment support measures, you can transfer any that exist into the fields INVSL01, INVSL02,... and so forth. The investment keys can be entered in these fields in any order. However, you must begin with the field INVSL01, and follow in an unbroken sequence.

### Completing the Blocks for Area-nn in the Transfer Structure BALTD for

## Rules for Setting Up the Input File

### Master Data

It is particularly important to make sure that you only transfer fields in the transfer record that are available for input during the running of the transfer program. For example: the depreciation area specified in AFABE02 copies its acquisition values from the depreciation area specified in AFABE01 (the option "Identical acquisition values" is set in the definition of the depreciation area). Then the corresponding field for the cumulative acquisition value is not available for input using the transaction AS91. That is, the field KANSW02 cannot be transferred (the symbol '/' is in the first position of the field KANSW02, see above). Therefore, we recommend that you look at one transfer transaction online for each company code, asset class and transaction. You can then see the fields that are ready for input in each case.

When structuring the transfer record, on the other hand, you can ignore the sequence of the depreciation areas on the depreciation area screen and value screen of the transfer transaction. For example, area 20 can stand in AFABE01, even if area 20 is the 4th area to be supplied with values in the value screen of transaction AS91.



The content of the field AFABE $nn$  may not differ from one company code or asset class to another! This is especially important in the following cases:

Within an asset class you can have assets both with and without investment support keys. If the investment area is specified in AFABE $nn$ , then AFABE $nn$  must remain the same, even for assets that do not have investment support keys, although nothing will be transferred into the other fields in the area- $nn$ . If, for example, the investment area 51 is transferred in BALTD-AFABE04, then the field AFABE04 = 51 must also be set for assets without investment subsidies. However, all other fields in the block for area- $nn$  should contain '/' in the first position.

The content of AFABE $nn$  also has to remain constant if you wish to change fields using AS92, even if there are no fields from area- $nn$  to be changed in the corresponding record.

In addition, you should observe the following rules when completing the blocks for area-01 to area-08:

If the field AFABE $nn$  = / is set, then the transfer program expects that no further fields are to be transferred for all following blocks for area- $mm$  (where  $mm \geq nn$ ).

*In other words:* The depreciation areas to be transferred must be transferred in a *consecutive*, unbroken sequence of blocks, starting with area-01. The contents of the fields AFABE $nn$  must also remain constant in this instance (per company code and asset class), even if certain depreciation areas are not transferred for certain assets.

## Batch Input and Direct Input

For the legacy data transfer of fixed assets, you have the Standard Functions of the Data Transfer Workbench at your disposal. The following contains information on the batch input and direct input procedures for legacy data transfer that are **specific to Asset Accounting**.

### Legacy Data Transfer Using Batch Input Procedure (Program RAALTD01)

When you use the batch input procedure, the system creates a batch input session containing the legacy data, and automatically processes this session. This method supplies the legacy data transaction of the *Asset Accounting (FI-AA)* component with data. The system writes any transfer records that have errors to an error session for later manual processing.

### Legacy Data Transfer Using Direct Data Input Procedure (Program RAALTD11)

When you use the direct data input procedure, the system writes the legacy transfer data directly to the appropriate tables in the *Asset Accounting* component. The system writes any assets that have errors to an error file. This procedure normally offers improved performance, as compared to the batch input procedure. However, there are certain limitations as regards the input data and errors. For more information, see [Direct Data Import \[Page 28\]](#).



When the number of legacy assets is fewer than 50 000 to 100 000, you should carry out legacy data transfer using batch input, since this procedure is better able to ensure that incorrect data is not transferred to the R/3 System. Depending on your hardware, the batch input interface is normally able to import approximately 150 asset master records per minute (including transactions) to the R/3 System.

For large numbers of legacy assets, SAP recommends the direct input procedure, since it can increase performance by about 10 times in comparison to transferring using batch input.

## Object-Specific Settings and Options

You use the activity category to choose either batch input transfer or direct data import.

### Constraints

- The Data Transfer Workbench limits the transfer structure to a maximum of 5 000 characters. Therefore, using the Data Transfer Workbench is possible only if you have fewer than 10 depreciation areas. When you have 10 or more depreciation areas, transfer structure BALTD (asset master data and values) can become longer than 5 000 characters. In that case, you have to carry out the transfer directly, either using report RAALTD01 (batch input) or RAALTD11 (direct data import) in transaction SE38.

You can use the Dictionary (transaction SE11) to find out the length of your BALTD transfer structure (choose *Dictionary object* → *Display*). To see a screen listing the header data and fields of the transfer structure, choose *Utilities* → *Runtime object* → *Display*. You see the character length of your transfer structure in the *Len* column in the header data.

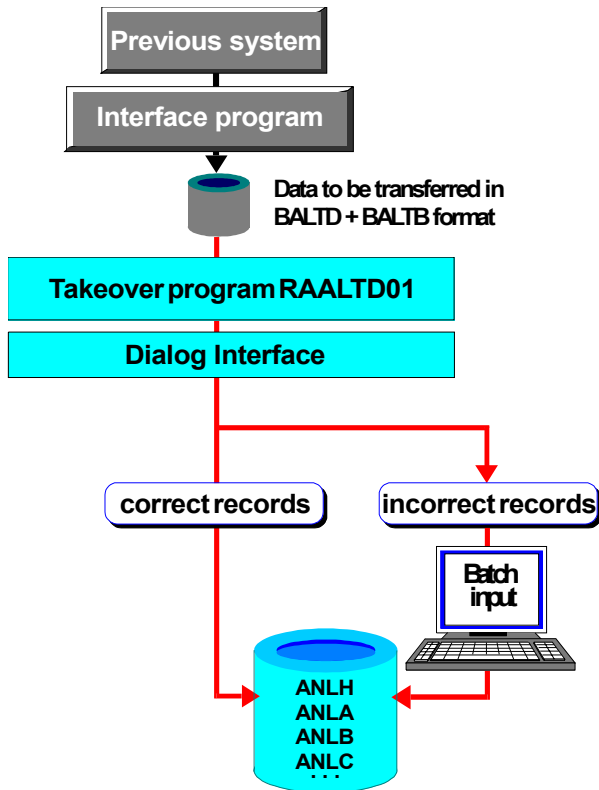
- Net book values cannot be transferred. You have to transfer gross asset values (APC and accumulated depreciation). The system then calculates the net book values.

**Batch Input and Direct Input**

## Legacy Data Transfer Using Batch Input

### Use

You can transfer asset master data and asset transactions into the FI-AA System using batch input (FI-AA Customizing: *Asset Data Transfer* → *Automatic Data Transfer*). The transfer program provides the transaction for asset data transfer (dialog interface) with the data from your legacy system. Incorrect data records are written to a separate batch input session. From there they can be manually corrected.



### Batch Input Legacy Data Transfer



If you expect performance problems due to large amounts of data, refer to Direct Data Import.

## Legacy Data Transfer Using Batch Input

### Error Correction for Batch Input

When you transfer legacy data using batch input, the transactions that are triggered are carried out in the background. If there are no errors found, posting takes place. If errors are found, the transactions with errors are placed in the batch input session under the name you entered in the *Name of the error session* field. By processing this batch input session, you can later correct the transactions with errors.

Statistics are created for every transfer run. In the case of errors, the first error that appears (for each asset) is logged. By selecting the logged error, you can display the long text for the error message.

#### Causes of Errors

Serious comprehensive errors that cause a program termination are:

- Company code is not in the input record
- Asset class is not in the input record
- Asset class requested is not defined in Customizing for *Asset Accounting*
- Number range is missing
- Transaction code is missing
- Unexpected record type (not A or B)
- *Number of transactions* field (BALTD-BWCNT) is not numerical
- Transactions with transaction code AS92 (change legacy data)

#### Errors that Cause Termination

If no database updates were made before the error occurred, the error causes a simple termination. Once the error is corrected, the transfer run can be restarted from the beginning.

If, however, assets were already processed and database updates carried out when the termination occurred, the system generates a log about the data records already processed, as well as the cause of the termination. You must now be aware of two things:

- **Error session**

It is possible that errors already occurred in records before the termination. These are now recorded in the error session and must be reprocessed.

- **Repeat run**

The key of the first data record that was not processed is displayed in the termination log. The cause of the termination has to be removed. You have to create an input file that contains the corrected data records, starting with (and including) the record at which termination occurred. After this, the program can process the newly created input file that contains only those records that were not yet processed.


As an alternate solution, you can reset the company code for Asset Accounting. If you reset the company code, you could then use the original input file with all of its data records.

In the test run, as well as in the update run, you can process the transactions in the foreground using the *Transaction flow in the foreground* option. In a test run, however, this only allows you to

**Legacy Data Transfer Using Batch Input**

observe **one** transaction. This occurs because no actual posting takes place, and the system therefore always stops on the last transaction screen.



To be able to see more transactions during a test run, you have to leave each transaction by choosing  *Cancel* when the transaction is finished (OK code field is blank). Choose *No* in the resulting dialog box.

## Legacy Data Transfer Using Batch Input

### Direct Input

For large numbers of legacy assets (refer to [Legacy Data Transfer \[Ext.\]](#)), you can choose the direct input procedure (direct data import) as an alternative to the batch input procedure. In contrast to batch input, direct data import writes legacy data directly to the respective tables. In comparison to the transfer using batch input, this increases performance by at least 10 times.

#### Constraints

Both procedures for legacy data transfer accept the same input structure. However, there are certain restrictions that apply to direct data import:

- The transfer date has to be the end of the previous fiscal year or come after the end of the previous fiscal year. In addition, a fiscal year change cannot have been carried out in the current fiscal year. This means that only the current fiscal year is open for posting in the R/3 System, but no previous or future fiscal years can be open.

**Example:** The current fiscal year is 1999. The transfer date has to be December 31, 1998 or a date in the year 1999. As long as you are still transferring legacy data, you cannot carry out a fiscal year change to the year 2000.

- All value fields are allowed to contain only one decimal point, and all blank spaces must be filled with zeros. This makes it possible to avoid time-consuming data checks and conversions.
- The depreciation start date must be entered.
- It is imperative to carry out a preliminary sort of the transfer records according to the key for input structure BALTD before a test run or an update run.

#### Consistency Checks

The following important consistency checks are carried out during direct data import:

- Valid capitalization date: The capitalization date has to be before the transfer date.
- The system checks for the existence of the asset class and the master data specifications related to account assignment (such as, cost center, business area).
- If the *Cost center* and *Business area* fields are defined as required fields in the screen layout, but no values are transferred for these fields, the program does not transfer these assets. Instead the program writes them to an error file (see below).

The screen layout control is taken into account dependent on the maintenance level. If there are fields that are suppressed according to the screen layout, but should nonetheless be supplied with contents from the input structure, this does not lead to an error. However, the system ignores these values. The asset is also transferred if there are conflicts with the default values from the asset class. Records from the input file are only rejected if records would be duplicated in the data base, or if there are obvious errors in the key in the structure of the record.



The direct data import also ensures that the essential checks for Asset Accounting are carried out. In comparison to the batch input procedure, the checks that are omitted in direct data import for performance reasons are relatively unimportant (for example, the existence of evaluation groups). It is particularly important to note that no loss of data can occur when you use direct data import.

### Legacy Data Transfer Using Batch Input

Therefore, it is recommended that you carry out sample checks of the input file using the batch input transfer procedure, to check for the correct record structure and logic. After this check, you can use direct data import.

You can program your own user-defined checks (such as for the validity of evaluation groups) using customer enhancement ALTD0001.

### Field Formats

When you use the direct data import, no conversion takes place; that is, the system accepts all input data without any further checks. The R/3 System normally converts entries for certain fields at the time of the online entry. During the direct data import, the system does **not** convert these entries as it does when you transfer using batch input. Therefore it is important that you supply the data in the correct format.

You can generate a list of all fields for which conversion takes place during online entry. Follow the procedure below:

1. Call the Data Browser (transaction SE16) and enter table DD03P.
2. Choose *Settings* → *Fields for selection*. Choose *CONVEXIT* in addition to the fields already selected.
3. Enter ANLA, ANLB, ANLC, ANLH, ANLZ as table names. For *CONVEXIT* select *not equal to* (in the dialog box that appears when you press selection options).
4. Choose *Execute*. You receive a list of all fields for which conversion is normally carried out.
5. Choose *Settings* → *List format* → *Choose fields*. Select the field descriptions *Table name*, *Field name*, *Language* and *Conv.routine*.

The list shows the fields that are converted when online input is used, and also shows the conversion routine. For the data import, you have to supply the fields in the target format of the conversion routine being used.

Important: The R/3 fields that use alpha conversion (such as cost center) have to be provided with leading zeros, if the strings being transferred are numerical only. For example, cost center 4711 has to be transferred as 0000004711.

### Format Restrictions

- The following restrictions apply to amount fields:
  - Amount fields can contain only the numbers 0 to 9, comma separator (for thousands), decimal point, and negative sign.
  - Always enter the decimal places.
  - Do not enter positive signs. Enter negative signs at the end of the number (on the far right).

You can generate a list of the amount fields by using the procedure above and making some changes. Use the selection option DATATYP = CURR. Remove the *CONVEXIT is not equal to blank* selection.

- The following restrictions apply to quantity units:

When entering quantity units, you have to use the internal format (for example, ST for unit). Do not use a language-specific name. You can generate a list of the internal quantity units using the Data Browser for table T006.

## Legacy Data Transfer Using Batch Input

### Refer to:

For general information on the direct input procedure, see *Cross Application Components → The Data Transfer Workbench → Data Transfer Techniques*.

## Error Correction for Direct Input

Along with assets that fail the consistency checks of legacy data transfer program, assets with the following general errors are rejected:

- Errors in the control tables, such as currencies that are not maintained
- Errors that occur during the calculation of depreciation
- Conflicts in number assignment
- General errors, such as incorrect record types, or if the company code or transaction code is blank or 0 in the input record.

The system lists all of these assets with errors in an error file. You enter the name of the file when you start the program. You have to manually work through the assets in the log. The system does not create a batch input session for assets with errors.

After all errors are corrected, you can run the RAALTD11 program again, using the error file as the input file. If it is not possible to find the cause of an error, then you can use the RAALTD01 to analyze the error file. Run the program in the foreground (choose *Transaction flow in the foreground*).