

BAAN IVc
Multisite

User Manual

U7031B US



Document information

Document

Document code : U7031B US
Document group : User Documentation
Document title : Multisite
Application/Package : BAAN IVc
Edition : B
Month modified : July, 1999

Check the BaanDoc database to see if this document is still a valid edition.

© Copyright 1999 Baan Development B.V. All rights reserved

The information in this document is subject to change without notice. No part of this document can be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Baan Development B.V.

Baan Development B.V. assumes no liability for any damages incurred, directly or indirectly, from any errors, omissions or discrepancies between the software and the information contained in this document.

Table of contents

Document information	i
About this document	v
1. Introduction to multisite	1-1
2. The company	2-1
2.1 Definitions	2-1
2.1.1 <i>Company</i>	2-1
2.1.2 <i>Logistics Company</i>	2-3
2.1.3 <i>Finance Company</i>	2-4
2.1.4 <i>Group Company</i>	2-5
2.1.5 <i>Base Company</i>	2-6
2.1.6 <i>Primary Company</i>	2-7
2.2 Company Structures	2-7
2.2.1 <i>Single Logistic / Single Finance</i>	2-8
2.2.2 <i>Multi Logistic / Single Finance</i>	2-9
2.2.3 <i>Single Logistic / Multi Finance</i>	2-9
2.2.4 <i>Multi Logistic / Multi Finance</i>	2-10
3. Finance	3-1
3.1 General Ledger	3-1
3.1.1 <i>Intercompany Transactions</i>	3-1
3.1.2 <i>Intergroup Transactions</i>	3-1
3.1.3 <i>Ledger Accounts and Dimensions</i>	3-1
3.1.4 <i>Journal Voucher Transactions</i>	3-2
3.1.5 <i>Ledger History Display and Reports</i>	3-2
3.1.6 <i>Financial Integration</i>	3-2
3.2 Accounts Payable	3-2
3.2.1 <i>Miscellaneous Invoices</i>	3-2
3.2.2 <i>Invoices for Matching</i>	3-2
3.2.3 <i>Open Entries</i>	3-3
3.3 Accounts Receivable	3-3
3.3.1 <i>Miscellaneous Invoices</i>	3-3
3.3.2 <i>Invoices from Sales Control (SLS)</i>	3-3
3.3.3 <i>Open Entries</i>	3-3
3.4 Cash Management	3-3
3.4.1 <i>Payments</i>	3-3
3.4.2 <i>Receipts</i>	3-4
3.5 Financial Statements	3-4
3.5.1 <i>Standard Statements</i>	3-4
3.5.2 <i>Consolidated Statements</i>	3-4
4. Distribution	4-1
4.1 Sales Order Management	4-1
4.1.1 <i>Order Handling</i>	4-1
4.1.2 <i>Customer Management</i>	4-3
4.2 Purchasing	4-4

4.2.1	<i>Purchasing Scenarios</i>	4-4
4.2.2	<i>Electronic Data Interchange (EDI)</i>	4-8
4.2.3	<i>Purchase Requirement Collection</i>	4-8
4.2.4	<i>Supplier Performance Analysis</i>	4-9
4.3	Costing & Transfer Pricing	4-9
4.3.1	<i>Costing</i>	4-9
4.3.2	<i>Transfer Pricing</i>	4-9
5.	Manufacturing	5-1
5.1	Product Definition	5-1
5.2	Planning and Forecasting	5-1
5.3	Manufacturing Planning	5-2
5.4	Distribution Planning	5-2
5.5	Subcontracting	5-3
5.6	Work Centers and Sub Work Centers	5-3
5.7	Enterprise Information System	5-4
6.	Common data	6-1
6.1	Customers and Suppliers	6-1
6.2	Logistic and Financial Tables	6-1
7.	General technical subjects	7-1
7.1	Terms and definitions	7-1
7.1.1	<i>Data Sharing</i>	7-1
7.1.2	<i>Data Replication</i>	7-1
7.1.3	<i>Logical Table Linking & Distributed Transactions</i>	7-2
7.1.4	<i>Exchange</i>	7-2
7.2	Table References	7-2
7.3	Networks	7-3
7.4	Client/Server Scenarios	7-4
7.5	Choosing a data sharing method	7-6
7.6	Introducing the case 'Remote Control'	7-7
8.	Sharing data	8-1
8.1	Multicompany Structures	8-1
8.2	Data Sharing by Package	8-2
8.3	Multiple Data Servers	8-2
9.	Asynchronous data replication	9-1
9.1	Introduction	9-1
9.2	Exchange	9-2
9.2.2	<i>Full Exchange</i>	9-4
9.2.3	<i>Audit Based Exchange</i>	9-6
9.3	Database generated replication	9-7
9.4	Independent data replication packages	9-10
10.	Synchronous transaction processing	10-1
10.1	Introduction	10-1
10.2	Logical Table Linking	10-2
10.3	Distributed transactions using the Baan B-shell	10-4
10.4	Distributed transactions by Baan's Database Driver	10-5

10.5	Distributed transactions using triggers	10-7
11.	Upgrading in multisite situations	11-1
12.	Multisite MPS on multiple servers	12-1
13.	Appendix A Sharing data in multicompany structures	13-1
13.1	Multicompany Environments	13-1
13.2	Multi Logistic/ Single Finance	13-1
13.3	Single Logistic/ Multi Finance	13-3
13.4	Multi Logistic/ Multi Finance	13-5
14.	Appendix B Datasharing by package	14-1
14.1	Common Data (TC)	14-1
14.2	Distribution (TD)	14-3
14.3	Manufacturing (TI)	14-9
14.4	Finance (TF)	14-13
15.	Appendix C In-doubt transactions	15-1
16.	Appendix D Reference counters	16-1
17.	Appendix E Hash columns	17-1
18.	Appendix F Input data for multisite MPS run	18-1
19.	Appendix G Company set up	19-1

About this document

This document is intended to provide functional and technical information about multisite concepts in Baan. During the implementation a multisite scenario must be defined. The multisite scenario that is chosen depends on many requirements which are discussed in this document including business processes, data management, and technical requirements.

Because each organization has unique requirements, this document does not attempt to determine the preferred multisite structure for a specific situation. Such a decision must be made after a thorough investigation of business and technical requirements, and Baan's functional and technical capabilities.

It is assumed that the reader has a basic understanding of the Baan software and is familiar with common Baan terminology. For a complete understanding of multisite in Baan, this document must be used as a supplement to the Multisite workshop.

This document is based on Baan IVc4.

Contents

Chapter 1 briefly introduces the multisite concept.

Baan's multisite functionality typically involves processing with more than one company. Throughout Baan the term **company** is used in several contexts. Chapter 2 provides definitions for all the uses of the term company, as well as examples of the four possible multicompany structures.

Chapters 3 to 6 discuss the multisite functionality of the Finance, Distribution, Manufacturing and Common packages.

Chapter 7 discusses several technical topics, including sharing data, client/server techniques and the related terms and definitions.

Chapter 8 discusses sharing data from the perspective of a multicompany structure and by package.

Chapter 9 deals with asynchronous replication of data. If asynchronous replication methods are used, both servers have a copy of the shared records. There are two or more instances of the same data record. The user always accesses its local database. Data is replicated from one table to the other.

Chapter 10 discusses synchronous transaction processing. In synchronous transaction processing, the user always accesses the actual data records. Each data record is only has one instance. During one transaction multiple records may be affected. If all data is located on one server, the data tables can be logically linked. If the data is located on separate servers, distributed data transactions can be used. For each of the data sharing methods a description is given and the possibilities and limitations are discussed.

Chapter 11 deals with the upgrading of functionality. When it is decided to upgrade the Baan applications it is unlikely that this can be done on all servers at exactly the same moment. As a result, for a limited period of time multiple versions of the application will have to communicate. This chapter briefly discusses the possible difficulties.

Chapter 12 deals with the working principles of multisite Master Production Schedule (MPS) functionality. It explains why multisite MPS functionality depends on the network structure and the number of servers.

Throughout the technical chapters a case is used to illustrate the characteristics of the data sharing methods.

Appendices A and B provide lists of the tables that must be and can be shared in the various multicompany structures. The remaining appendices briefly discuss various multisite related topics.

Acronyms used in this document

Acronym	Full term
ATP	Available to Promise
DRP	Distribution Requirement Planning
EDI	Electronic Data Interchange
EIS	Enterprise Information System
IRP	Intelligent Resource Planning
LAN	Local Area Network
MPS	Master Production Schedule
MRP	Material Resource Planning
SIC	Statistical Inventory Control
WAN	Wide Area Network

1.

BAAN IVc

The term multisite can be used to describe **processing** in more than one **business unit** within an organizational structure. To avoid confusion two words from the previous sentence must be described:

- 1 Processing: This can be actual business events, i.e. material handling, manufacturing, administration; as well, it can be data processing, .i.e. recording of these business events.
- 2 Business Unit: Any entity within an organization that has some degree of independence, i.e. warehouse, distribution center, manufacturing plant, sales office, administration group, etc. Independence implies that some degree of management is unique to each entity.

Within Baan, the term multisite can apply to a broad range of topics which includes both functional and technical concepts. Some examples of multisite concepts within Baan are group company structures, companies, warehouses, dimensions, series, data sharing, client/server, etc.

It is obvious that there are many ways to interpret the term multisite, both within an organization, and within Baan. During implementation the relationship between the organizational structure and the Baan company structure is defined.

The following example demonstrates how organizational structures can be mapped to Baan company structures:

- One physical building houses two separate production lines for two separate product lines. Both product lines share a large stock room that resides in this same building.
- There are several ways this organizational structure can be mapped to Baan companies:
 - Within one company use two ranges of work centers which represent the two production lines. Use two warehouses to represent the stock belonging to the two product lines. Use dimensions to group the production and inventory for financial reporting.
 - Within two companies, use each company for production and warehousing of the product lines. Use only one finance company, using dimensions to group the production and inventory for financial reporting.
 - Within two companies, use each company for production and warehousing of the product lines. Use both finance companies to do financial reporting.

This simplified example demonstrates that there can be several ways that organizational structures can be mapped to Baan company structures.

The company structure defined during implementation will depend on many factors, including physical location, technical architecture and data management requirements, management structure, reporting requirements and centralization/decentralization of functions such as planning, procurement, manufacturing and financial administration.

The process for mapping the structure involves analyzing the organizational structure and processing requirements. These requirements are then reviewed to determine how they can be met using Baan's multisite functional and technical capabilities.

The remaining chapters discuss many critical aspects of a multisite implementation.

2.

BAAN IVc

2.1

Definitions

There are several types of companies and company structures available, which can or can not be related, share data, and/or process transactions among themselves. The Manufacturing, Distribution, Finance and Common chapters in this document describe how companies can interact with each other using the multisite functionality.

This chapter provides definitions for the following terms:

- Company, including company zero
- Logistics company
- Finance company, including group company, base company, and primary company

This chapter also describes the four company structures:

- Single logistic / single finance
- Multilogistic / single finance
- Single logistic / multifinance
- Multilogistic / multifinance

2.1.1

Company

Within the Baan software companies are used to reflect the structure of an organization. There are two important aspects of a company: data and functionality.

- **Data:** A company consists of a set of tables accessible to users working in that company, i.e. customers, items, ledger accounts and orders.
- **Functionality:** A company is an entity used to process and manage business transactions, i.e. purchase, sales and production orders, including the related financial transactions.

Every company is divided into two main components: logistics and finance, with each having related data and functionality. This is illustrated in Figure 1. Refer to the paragraphs following for definitions of logistics and finance companies.

Companies can be numbered from 001 to 999.

Company Zero is a separate company that contains technical software configuration data used by all companies operating within the same Baan installation. Company Zero is provided as part of the installation of the Baan software. The Baan Tools package consists of tables that belong to Company Zero (000).

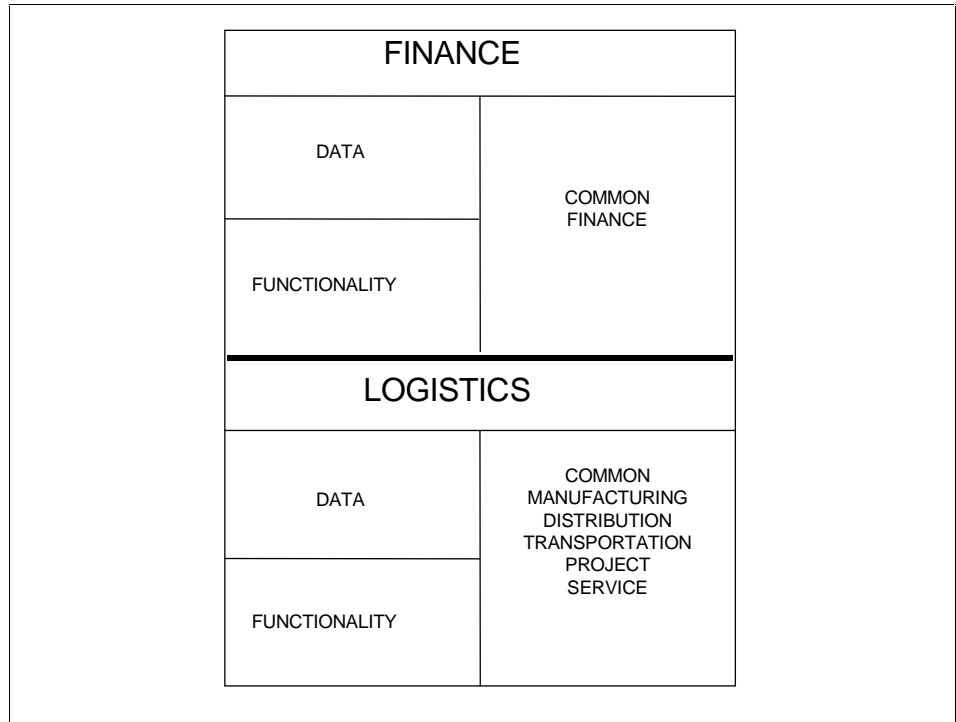


Figure 1 Components of a company

2.1.2

Logistics Company

The term logistics company is used to describe the logistics component of a company that is used for operational processing. The logistics component consists of data and functionality for the packages:

- Common
- Distribution
- Manufacturing
- Transportation
- Service
- Project

There is no hierarchy among logistic companies.

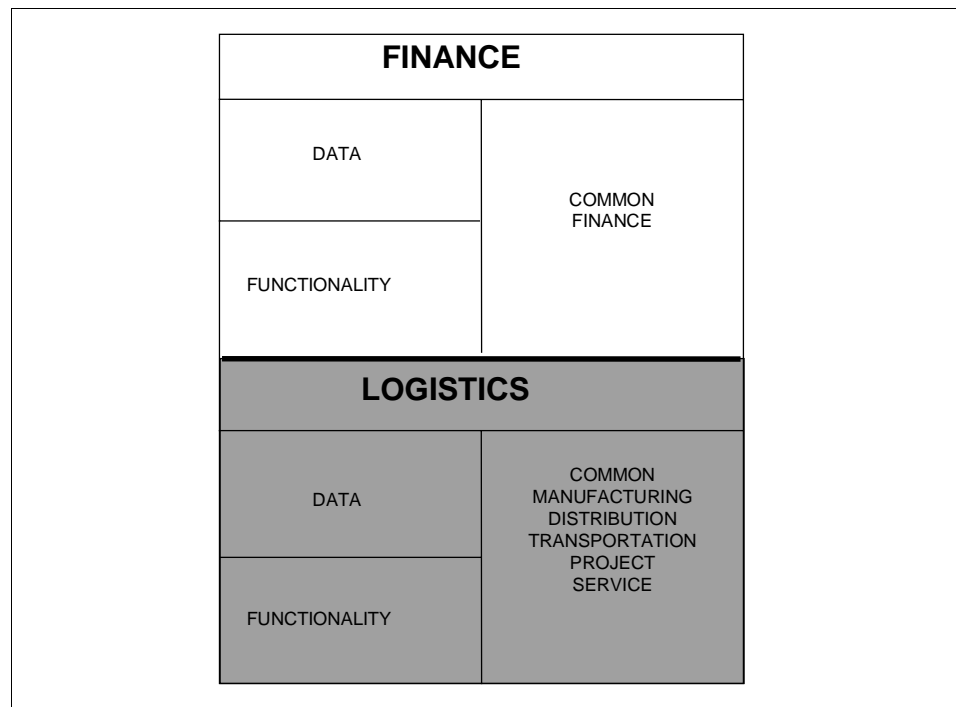


Figure 2 The logistics components of a company

2.1.3 Finance Company

The term finance company is used to describe the finance component of a company that is used for financial processing. The finance component consists of data and functionality for the packages:

- Common
- Finance

The only hierarchy among finance companies is via the group company designation of one of the finance companies.

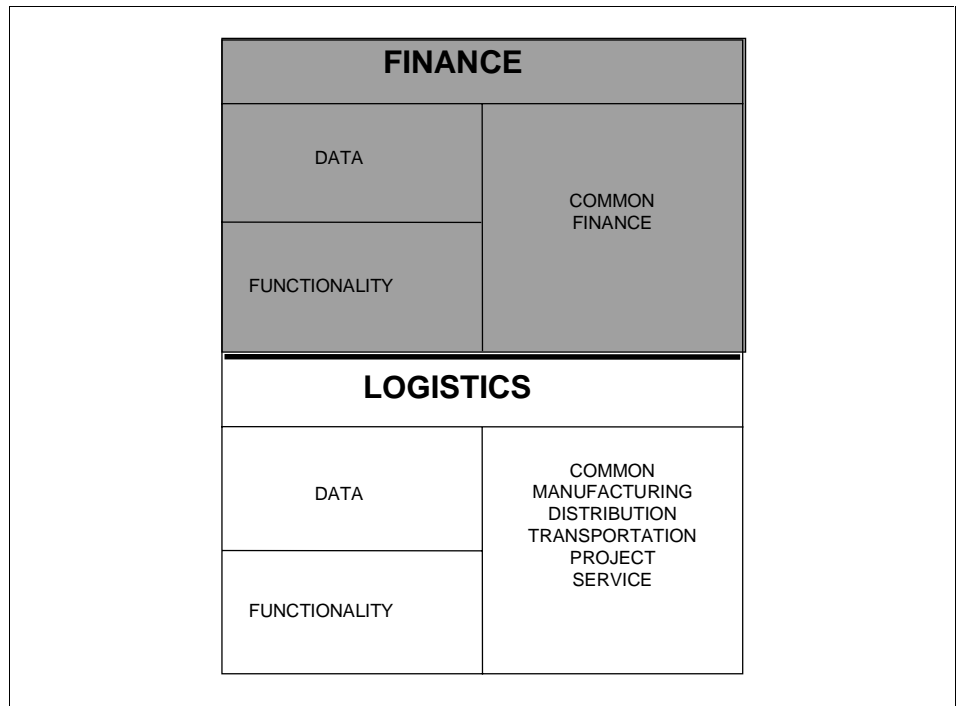


Figure 3 The finance components of a company

2.1.4

Group Company

In an organization that consists of more than one company, specific functionality is available for processing between companies. The designation Group Company is given to a specific company within a group of related companies.

Within a group of companies, intercompany financial transactions (resulting from operational and/or financial processing) can be automatically generated. In addition, financial data for all companies in the group is available for inquiry and reporting purposes.

The characteristics for a group of companies are a common home currency, country and financial periods. A group company structure can include more than one legal entity and an unlimited number of companies.

The group company functionality provides a two level company hierarchy structure within the group, as financial data for all companies within the group is summarized in the group company. There is no hierarchy among group companies.

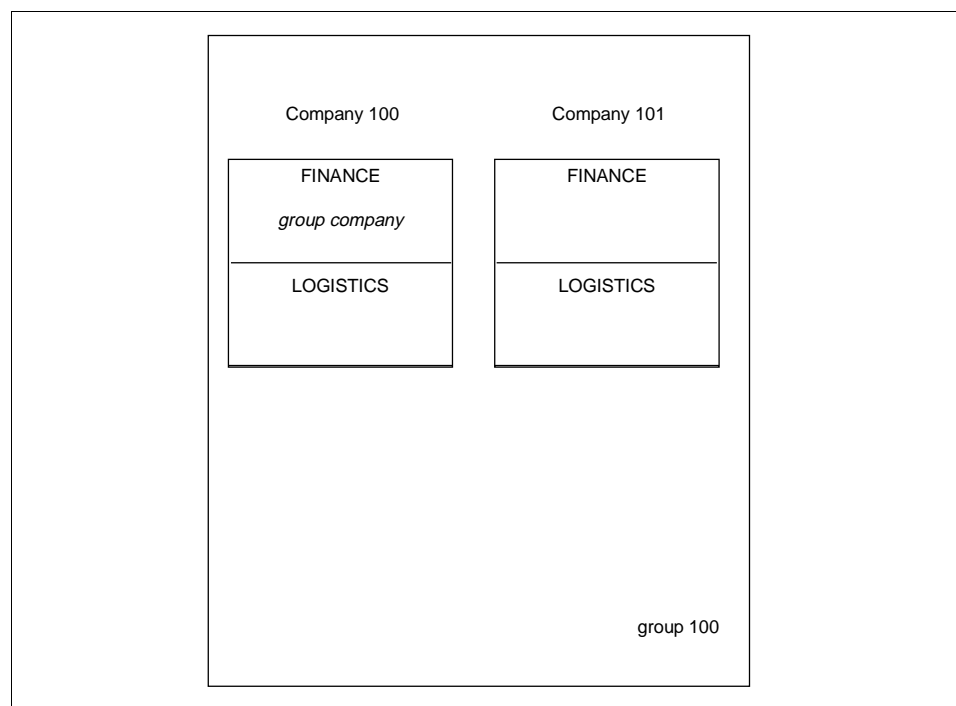


Figure 4 A group company structure

2.1.5 Base Company

In an organization that consists of more than one company, specific functionality is available for processing between groups of companies. The designation Base Company is given to one of the group companies.

Where more than one group of companies transact journal voucher entries, the company designated as the base company is used to manage the transaction processing between the groups.

In the example in Figure 5, either company 100 or company 200 could be designated as the base company to allow journal voucher transactions to take place between the two groups of companies.

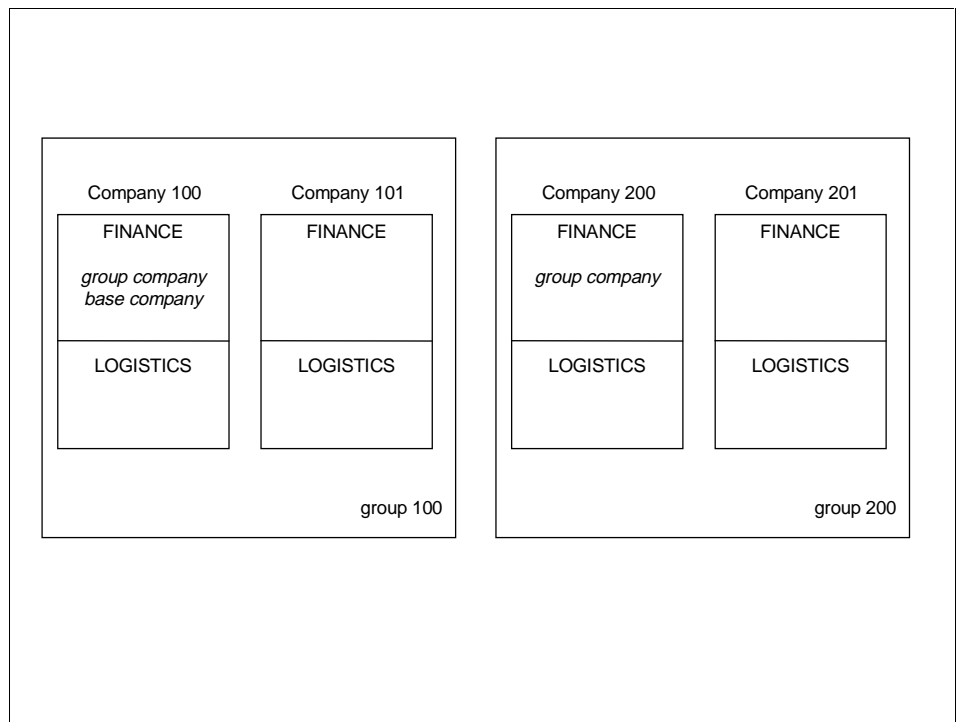


Figure 5 The base company in a multi group structure

2.1.6

Primary Company

Where more than one database server is used for a group company structure within a specific hardware environment, primary companies must be designated to allow group company functionality.

The primary company is a technical designation and is used to manage transactions within a group of companies, in the same manner as the group company does in a single server environment. The primary company designation does not replace the group company designation or its functionality.

The server that the group company resides on does not require a primary company, however, all other servers must have a designated primary company.

In the example in Figure 6, where database server 1 holds the group company (100), company 102 or 103 on server 2 would be designated as the primary company.

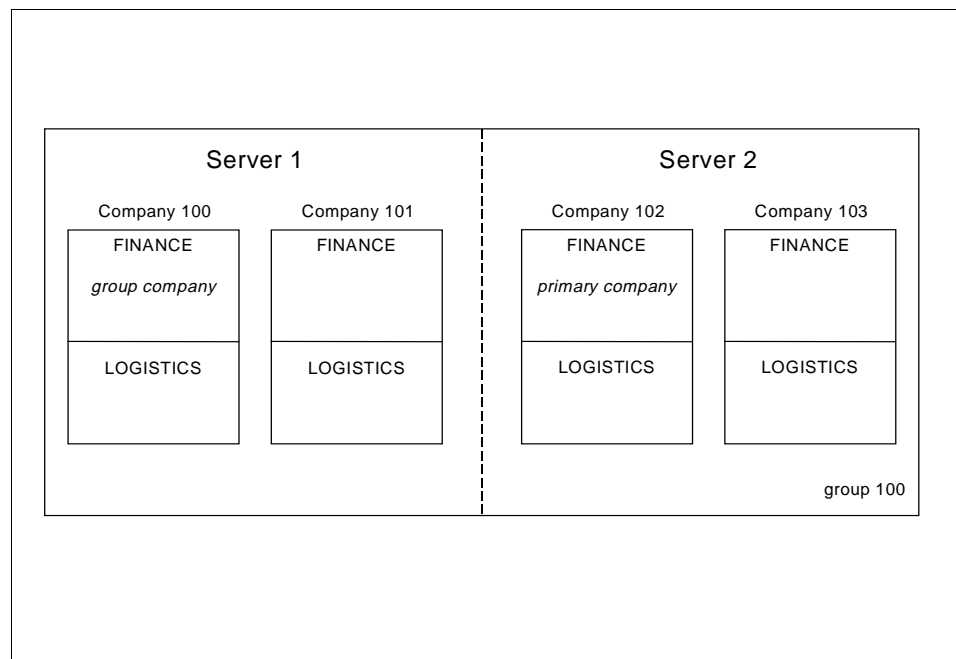


Figure 6 The primary company in a group company / multi database server environment

2.2

Company Structures

The following structures are available for companies within a group. The structure chosen during implementation will depend on the process, management and reporting requirements of the specific organization.

Any company within a group company structure can interact with any company in another group company structure using specific multisite functionality, i.e. MPS, EDI. In addition, certain data can be shared by companies within a group and across groups (refer to the Technical Reference Manual for further details).

The restrictions for all companies within a group are:

- one common home currency
- one common home country
- common financial periods

Within any company structure session, table and field level security can be used to restrict user access to data.

2.2.1 Single Logistic / Single Finance

In a single logistic / single finance (stand alone) structure all processing is done within one company, therefore only one set of tables is used. This restricts the organization to using only one standard cost per item and one bill of material per item.

Two or more single company structures can use EDI and multisite MPS to process between the companies. In addition the companies can share certain types of data.

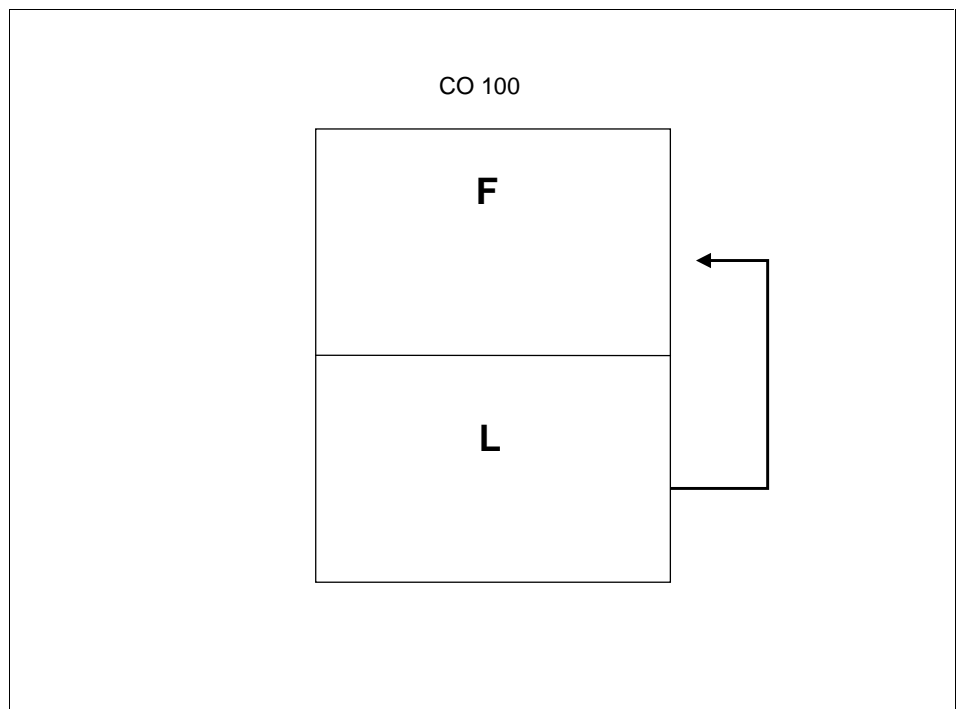


Figure 7 Single logistic / single finance structure

2.2.2

Multi Logistic / Single Finance

In a multi logistic / single finance structure the manufacturing and distribution processing can be done in several companies, while the financial processing is done in only one company. In this structure financial integration transactions from all logistic companies will be posted to only one finance company, therefore no intercompany transactions occur. The group company is the finance company used.

Two or more companies, within the group or across groups can use EDI and multisite MPS to process between companies, as well, the companies must share certain types of data.

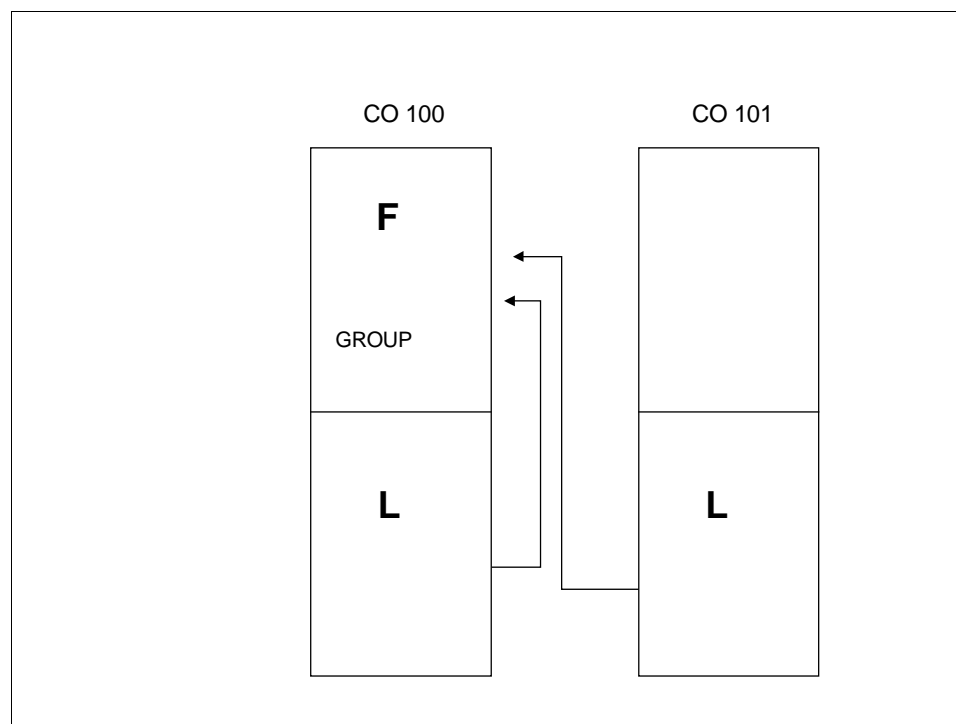


Figure 8 Multi logistics / single finance structure

2.2.3

Single Logistic / Multi Finance

In a single logistic / multi finance structure the manufacturing and distribution processing is done in one company, while the financial processing can be done in several companies. The logistic company number used is the group company.

In this structure financial integration transactions from the logistic company will be posted to the finance companies specified on the transaction. Intercompany transactions will occur when the finance companies on the debit and credit entries of the transaction are different. The finance company can be specified on the warehouse, work center, project, as well as on sales, purchase and production orders.

Two or more companies, each from different groups, can use EDI and multisite MPS to process between companies. Within a group of companies certain data must be shared, while across groups certain data can be shared.

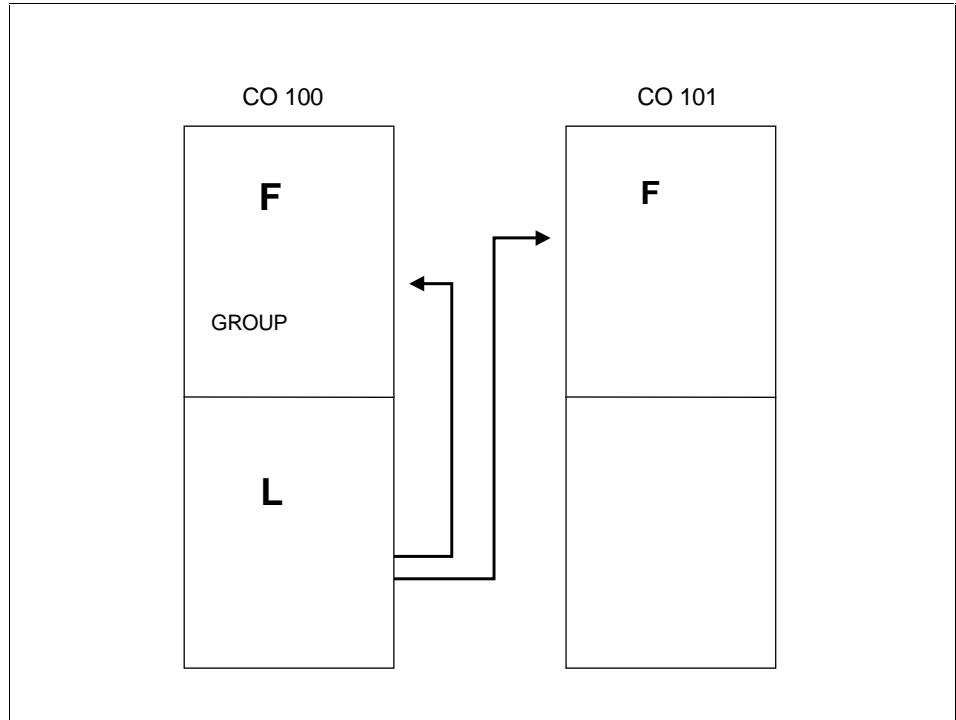


Figure 9 Single logistic / multi finance structure

2.2.4

Multi Logistic / Multi Finance

In a multi logistic / multi finance structure the manufacturing and distribution processing can be done within several companies, while the financial processing can be done in several companies. One of the companies (typically the company where corporate and administrative entries are processed), is designated as the group company.

This type of structure can include aspects of all the other types. That is, within one group company, one company can process as a single logistic / single finance structure, and several other companies can process as single logistic / multi finance and/or multi logistic / single finance structures. Any combination of these is possible.

In this structure financial integration transactions from the logistic company will be posted to the finance companies specified on the transaction. Intercompany transactions will occur when the finance companies on the debit and credit entries of the transaction are different. The finance company can be specified on the warehouse, work center, project, as well as on the sales, purchase and production orders.

When two or more companies within a multi logistic / multi finance structure process as a multi logistic / single finance structure, intercompany transactions will occur when the finance companies specified on the debit and credit entries of the transaction are different. Note that in a multi logistic / single finance structure that is not embedded within a multi logistic / multi finance structure, intercompany transactions are not generated.

Two or more companies within the group, or across groups, can use EDI and multisite MPS to process between companies. Within a group of companies certain data must be shared, while across groups certain data can be shared.

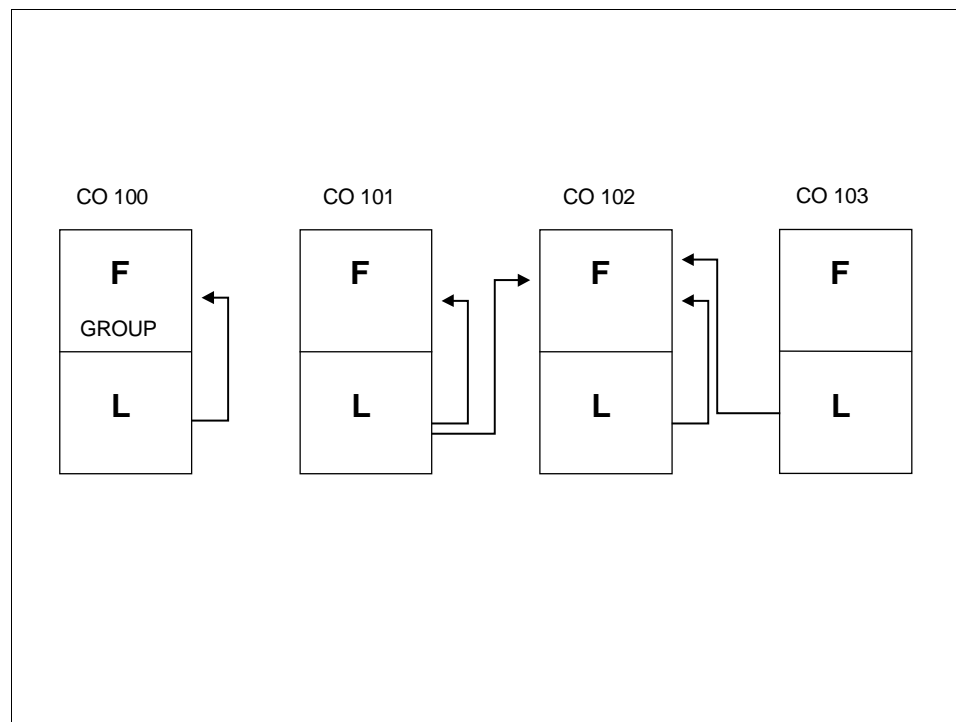


Figure 10 Multi logistic / multi finance structure

3. BAAN IVc

3.1 General Ledger

3.1.1 Intercompany Transactions

Within a group company structure, intercompany transactions are created when a financial transaction affects two or more companies. Because financial transactions within a company must balance, off-setting debit and credit entries are created automatically. The off-setting entries are created based on intercompany relations, which are user defined during implementation.

Intercompany transactions can result from transactions originating in finance, such as journal vouchers, accounts payable and receivable invoices, and cash transactions.

Intercompany transactions can also result from logistics operations. Specific logistic elements have a finance company designation, i.e. warehouse, work center (refer to the manufacturing and distribution chapters for the source of these transactions). Most integration transactions can have their debit and credit entries belong to different companies. For example, a production order in company 100 uses inventory from a warehouse with company 101 as its finance company.

3.1.2 Intergroup Transactions

Financial entries can be created affecting two group companies. The intergroup transactions are the result of journal voucher postings within a group company that affect specific ledger accounts. The entries for the other group company are created based on intergroup relations, which are user defined during implementation.

3.1.3 Ledger Accounts and Dimensions

Within and across companies, ledger accounts and/or dimensions can be grouped to represent business organizations.

Parent/child relationships and sub-levels provide roll-up and drill down capability within the ledger account and dimension structures.

Inquiry and reporting functionality within the finance package use the ledger account and dimension history to support multicompany visibility.

3.1.4 Journal Voucher Transactions

A journal voucher transaction can affect more than one company within a group company structure. Intercompany transactions will be created automatically based on user defined intercompany relations.

3.1.5 Ledger History Display and Reports

Within a group company structure multicurrency reports and displays of financial data are available.

Inquiry capabilities include group ledger account and dimension balances, with drill downs to company totals and ledger transaction detail (including integration data such as order number, customer or supplier number and item code).

Trial balances can be printed for the group company structure, and ledger account ranges can be selected.

3.1.6 Financial Integration

Financial integration set up data can be centralized. This means that for each type of integration transaction and logistic element combination, there can only be one definition of ledger account, dimension, transaction type and document series. For example, production issues from warehouse ABC can only integrate to one ledger account.

3.2 Accounts Payable

3.2.1 Miscellaneous Invoices

Both the debit and credit entries of an invoice can belong to any company within the group company structure. Intercompany transactions are created automatically, based on the user defined intercompany relations.

3.2.2 Invoices for Matching

Invoices can be entered in one company and matched to purchase receipts from any company within the group company structure. This centralized invoice entry will automatically create intercompany transactions (at approval), based on user defined intercompany relations.

Note:

Intercompany transactions will be created at the receipt transaction, based on user defined intercompany relations), if the finance company on the purchase order header is different from the finance company on the warehouse of the

purchase order line. In this case, no intercompany transactions are created if the invoice is entered in the same company that is on the purchase order header.

3.2.3 Open Entries

Central invoice management is possible for invoices entered in any company within the group company structure. Invoices can be printed and displayed for all companies within the group.

3.3 Accounts Receivable

3.3.1 Miscellaneous Invoices

Both the debit and credit entries of an invoice can belong to any company within the group company structure. Intercompany transactions are created automatically, based on user defined intercompany relations.

3.3.2 Invoices from Sales Control (SLS)

Accounts receivable invoices are created after sales invoices are printed in the Sales Control module. Intercompany transactions are never created from SLS sales invoices. Intercompany transactions are created from the sales issue transaction if the finance company on the sales order header is different from the finance company of the warehouse on the sales order line.

3.3.3 Open Entries

Central invoice management is possible for invoices entered in any company within the group company structure. Invoices can be printed and displayed for all companies within the group.

3.4 Cash Management

3.4.1 Payments

Payments can be created in the group company for invoices from all companies in the group company structure. Intercompany transactions, based on user defined intercompany relations, are created automatically when the payment company is different from the invoice company.

3.4.2 Receipts

Receipts can be entered in any company in the group company structure. Intercompany transactions (based on user defined intercompany relations), are created automatically when the receipt company is different from the invoice company.

3.5 Financial Statements

3.5.1 Standard Statements

Financial statements are user defined reports of financial history data. Each statement can include data from any company, ledger account and dimension within the group company structure.

3.5.2 Consolidated Statements

Consolidated statements combine data from standard statements for any company within an operating environment. Consolidated statements are not restricted to companies within a group company structure, and can include data from more than one group.

Column and line level conversion factors can be applied to translate data from different companies to a common currency or to reflect partial ownership of a company.

4.

BAAN IVc

4.1

Sales Order Management

4.1.1

Order Handling

Stock Visibility

During sales order entry the checking of available stock is important. Two types of stock can be distinguished: the stock owned by the company creating the sales order or the stock of other (sister) companies. The company owned stock is directly visible by viewing stocks within all warehouses of the related company. The allocation of this stock is simply performed by selecting one of the warehouses on the sales order line.

Next to the company owned stock, Baan IV also offers the stock visibility of related (sister) companies. The user can create a so-called Bill of Enterprise which specifies all warehouses (by priority) which are required on the display across companies.

The allocation of stocks that are *not* owned by the company creating the sales order happens in the following way. Since those stocks belong to another company the direct allocation of these goods by means of selecting a warehouse of that company is not possible. The goods will have to be purchased from that company in order to sell them to the customer. The *shipping* of the goods, though, can happen in different ways, as will be described further on. Another option obviously is to sell the goods from the other company, which takes away the need for a buy-sell transaction.

During sales order entry the system can perform an automatic stock check. This check is related to the warehouse selected on the sales order line. Next to that, the system can perform an Available-to-promise (ATP) check if the item sold is an MPS item. This check is currently only for the company entering the order. A Multicompany ATP check will be available within the new Intelligent Resource Planning (IRP) module.

Delivery Options

During Sales Order entry the user has various options to select from when deciding how the goods will have to be shipped to the final customer.

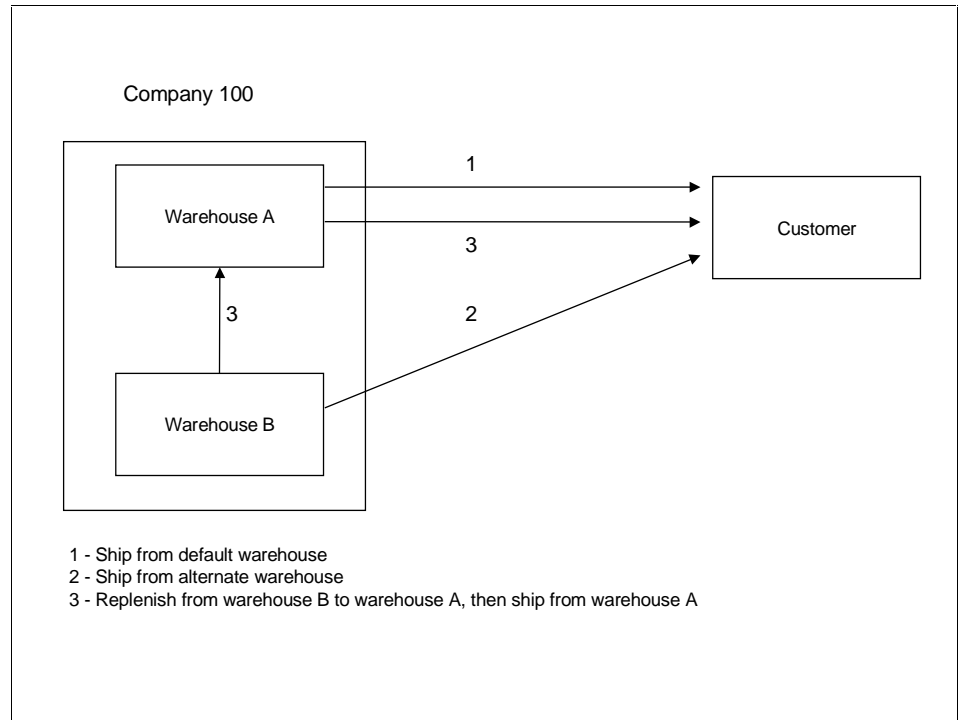


Figure 11 Delivery options 1

The goods can be shipped from the warehouse that defaults into the sales order (refer to 1 in Figure 11 above), or the sales order can be changed to ship from another warehouse in the same company (refer to 2 in Figure 11 above).

Replenishment orders can be used to transfer stock between two warehouses in the same company (refer to 3 in Figure 11 above). The replenishment order can be generated directly from order-entry, which will be linked to the original sales order. The replenishment order functionality allows a transfer of goods between two warehouses without having to invoice. Financially, the flow of goods will be registered between the two warehouses.

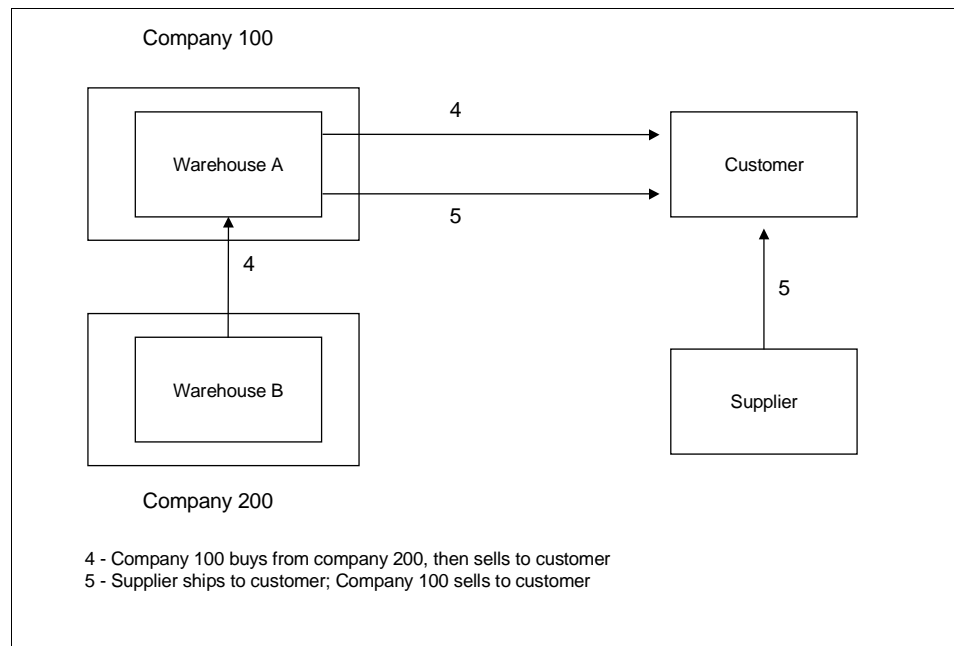


Figure 12 Delivery options 2

In the situation where the goods are supplied by an outside source (for example a sister company), a purchase order will be created, and can be linked to the original sales order (refer to 4 in Figure 12). With this option, the goods are shipped to the buying company, who will ship to the customer.

If the shipment to the final customer is done directly by the supplier (option 5), the *direct delivery* procedure is applicable (refer to 5 in Figure 12). The direct delivery functionality automatically creates a purchase order to a supplier, with the customer's address as the delivery address on the purchase order. The "supplier" could also be another Baan company.

Both with option 4 and 5 invoicing will take place between the selling and buying entities. These buy/sell procedures can be facilitated by making use of the EDI functionality, which will be discussed further on.

4.1.2

Customer Management

Customer File

In a multicompany situation a common requirement is to control the customer file from a central point of view. This file contains static data, like customer name, address, as well as dynamic data, like open balances. Depending on the situation, both static and dynamic data could require a centralized coordination. An example is the checking of the customer credit limit against the open balance. In Baan IV the synchronization of the customer file across multiple companies can be realized by the use of the Exchange module or the linking of tables. For more information about the use of these techniques the reader is referred to the technical document.

Customer Budgets / Sales Statistics

In the Sales Budgets module within Distribution, sales budgets can be specified in very flexible ways (e.g. by customer, region, item group, sales rep, etc.). Budgets can be aggregated from specific/detailed levels to more generic levels. To be able to aggregate budgets across companies the Exchange module can be used. The same statements can be made about sales statistics.

Sales Pricing

Sales prices are by default defined at the item level (read: at company level). This implies that price setting by warehouse, for example, is not possible.

Next to the default price, pricing agreements and sales contracts can be specified. This functionality is offered at company level. To synchronize these agreements/contracts across multiple companies the Exchange module can be used.

Performance Indicators

The Enterprise Performance Manager within Baan IV provides functionality to control the business based upon performance indicators. These indicators can be built up from data derived from multiple companies. An example could be to track delivery performance across sites.

4.2

Purchasing

4.2.1

Purchasing Scenarios

In a Multicompany environment there can be a need for central coordination of purchasing activities. These activities might, for example, encompass a central procedure for contract negotiation with outside suppliers or, alternatively, the central receipt of goods which subsequently need to be distributed internally. These examples illustrate the fact that central purchasing can have different shapes, depending on *what* is coordinated from a central point of view. This paragraph describes possible purchasing scenarios.

Scenario I: Local Purchasing / Central Contracting

In this scenario each site (read: company) has the local authority to purchase goods directly from an outside supplier. This implies that the local site will create purchase orders directly without involvement of a central purchasing department. The handling of goods receipt will be handled locally by the site as well.

The central coordination in this scenario consists out of the contracting and pricing with external suppliers. This process is performed by a central purchasing department. The local sites will make use of these contract conditions if local purchase orders are created. These orders will therefore carry the prices and discounts from a centrally negotiated contract.

From the point of view of the central department, the result of the local purchasing activities will be made visible on the central contract.

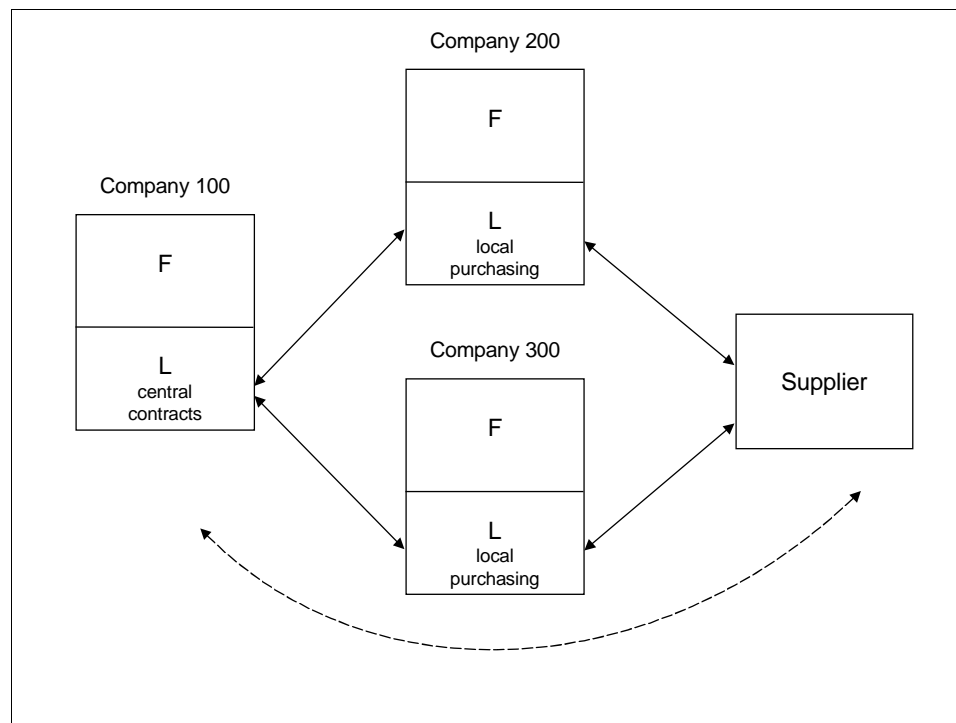


Figure 13 Local purchasing with central contracting

The situation illustrated by this picture can be modeled in the Baan package by using three 'companies'. Two of them are the local sites, the third company is the 'central contracts company'. To achieve the central contracts functionality, the contract tables must be synchronized. This can be done in different ways.

One way is to logically link these tables. This requires all companies to be stored on one database server (see: technical document). At the time of creating a purchase order locally, the price/discount from the contract will be applied automatically, and the contract will be updated with the 'called off quantity'.

The other option is to use the Exchange module for replicating the contract information.

Some additional remarks related to this scenario:

- 1 Triton 3.1 does not provide the functionality for analyzing the contract history for multiple companies from a single company, due to the fact that the company number is not stored in the history table. (This problem has been solved in Baan IVb)

- 2 If contracts are created both in the central company as well as in the local companies, ranges of contract numbers must be used.
- 3 Delivery schedules must only be used in the central contracts company for displaying information and not to update the planned inventory transactions.

Scenario II: Central Purchasing

In this scenario all purchase orders to *external* vendors will be sent by the central purchasing company. The local sites will place their requirements on this central company. The handling of goods receipt will be done centrally by the central company. Subsequently, the goods will be distributed to the local sites.

The contract and pricing negotiations will be performed centrally. Only the central company carries the central contract, since all external purchasing activities are performed only from this company.

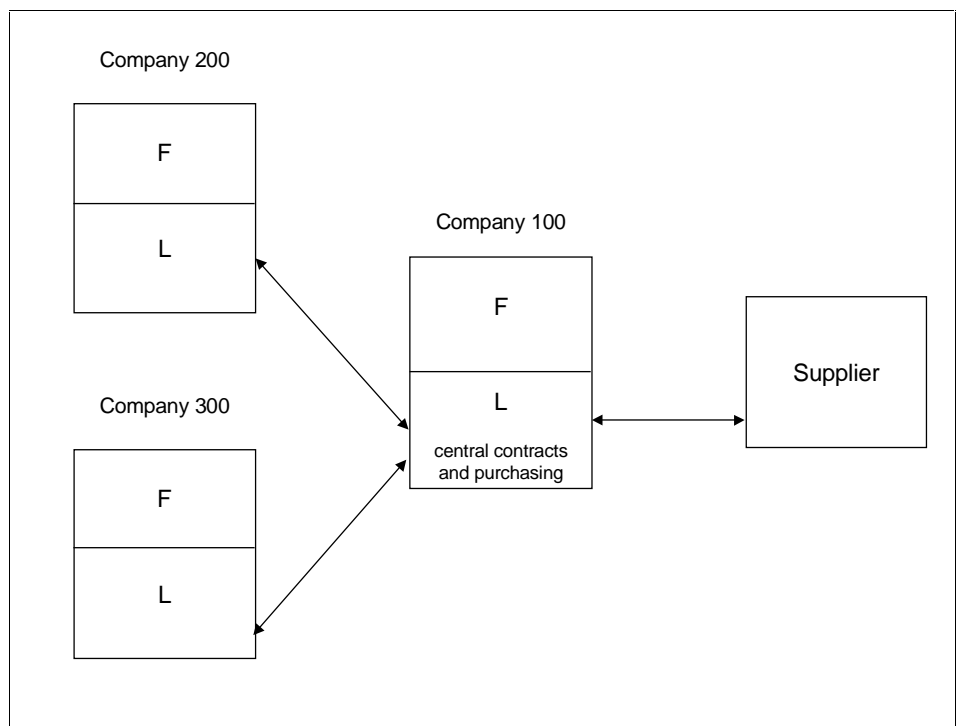


Figure 14 Central purchasing

This illustration can be modeled in Baan again by using three ‘companies’. The contracts will only be stored in the central company, since only this company will perform external purchasing. This implies that all vendor information (contract history, purchase history, supplier reliability, etc.) will, *by definition*, be visible in an aggregated format (no synchronization tools need to be applied), which is an advantage compared to scenario I. The disadvantage of scenario II is the overhead of handling the internal orders.

Scenario III: Central Purchasing, direct deliveries

This scenario is quite similar to the previous scenario. The main difference is the fact that delivery from the external supplier is done directly to the local sites. The central company therefore doesn't perform the central receiving function.

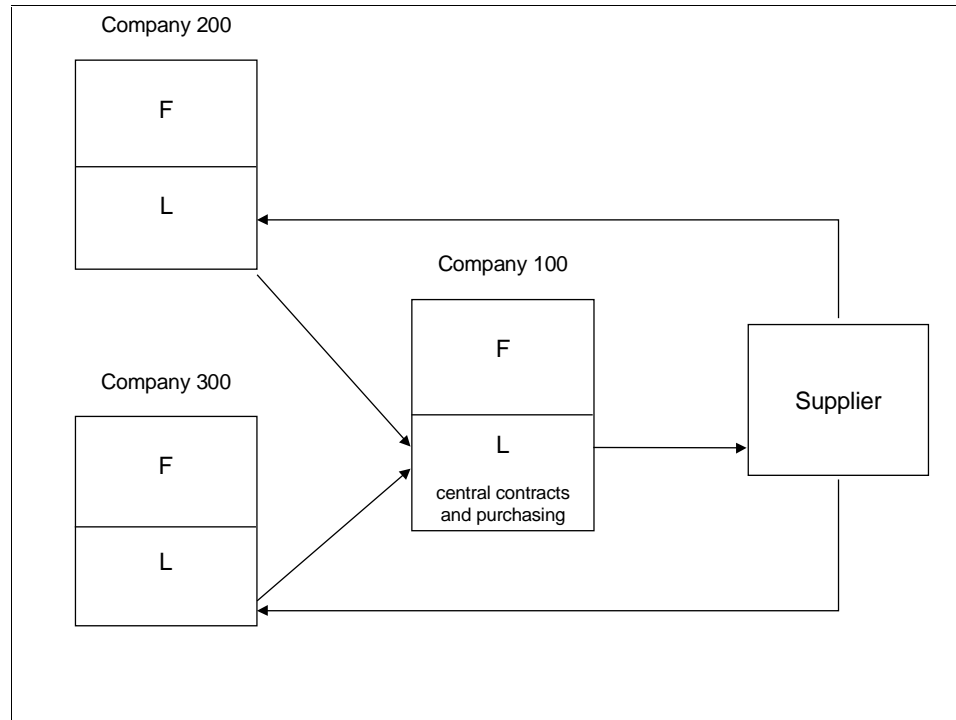


Figure 15 Central purchasing with direct deliveries

For *every* incoming order, in the central company, an external purchase order will be created. This comes from the fact that the direct deliveries require a one-to-one relationship between the orders. The implication is that the local purchase requirements will not be aggregated into one single external order by supplier. Due to this one-to-one relation, no planning needs to take place in the central company. This company basically 'pushes' the requirements through to the vendors.

With regards to the vendor information, the statement from scenario II applies to this scenario as well. All external purchasing information is only stored in the central company.

The advantage of this scenario compared to the previous one, is the fact that the overhead is only related to the handling orders and not to physically handling central receiving of goods.

4.2.2 Electronic Data Interchange (EDI)

The scenarios described above mention the use of intercompany orders; in scenario II and III the local company is purchasing from the central purchasing company. These purchase orders in the local companies will generate sales orders within the central purchasing company. This process is supported by the EDI module.

This module allows for the automatic generation and control of outgoing and incoming order messages.

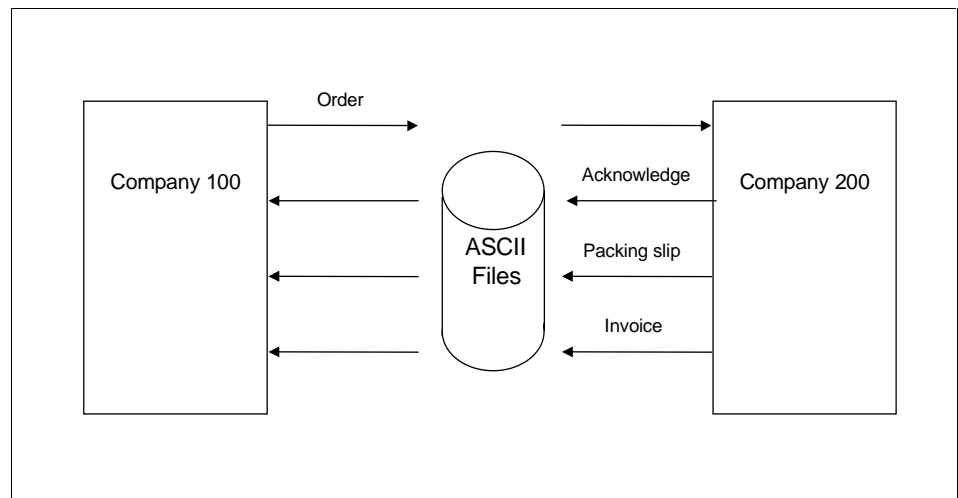


Figure 16 EDI

The EDI functionality currently does not support the handling of order changes. These changes have to be applied manually.

4.2.3 Purchase Requirement Collection

In the process of creating contracts and pricing conditions, the central purchase department needs a visibility of purchase requirements for the medium and long term.

Currently the collection of requirements *across multiple companies* can be realized by applying the Multisite MPS functionality. This requires that the purchase items are declared and controlled as MPS items. Since the MRP functionality only works at company level, all MRP purchase requirements need to be collected in a separate way. These requirements can be aggregated by using Exchange functionality.

Alternatively, purchase budgets can be used and aggregated as well by Exchange.

4.2.4 Supplier Performance Analysis

Tracking supplier performance with regards to reliability in terms of delivery times is possible via the supplier performance functionality of Baan. This functionality works at company level, and therefore a data synchronization mechanism like Exchange needs to be used to aggregated multicompany data.

Alternatively, the Enterprise Performance Manager can display performance indicators (PI's) based upon data from multiple companies. Using delivery performance PI's, supplier tracking can be established across sites. These PI's can be related to delivery performance regarding time and quality of the supplier deliveries.

4.3 Costing & Transfer Pricing

4.3.1 Costing

Companies operating in a multisite environment sometimes produce products in different ways, and at different locations. One product can be produced according to separate production plans making use of different routings, different bills of material. Obviously, these differences need to be reflected in the different planning and costing for the same item.

To calculate the standard cost of an item, Baan makes use of a so-called default routing. Based upon this routing and the bill of material of the item, a standard cost price is calculated. This cost price is for example used for updating the item stock value after stock transactions.

Having different cost prices for the same item, implies that stock transactions will have different impacts on the stock value, depending on which cost is used.

In the Baan package the issue of producing the same items against different routings, bills and costs is resolved by modeling the those production areas as separate company numbers. The Baan data model currently only supports one default routing and one standard cost price by standard item within one company number.

By making use of the Project (PCS) module of Baan, items can have costing as well as planning by project. The stock valuation in that case is handled by project.

4.3.2 Transfer Pricing

In a multisite environment goods are often shipped between sites. Sometimes a sales margin is added by the shipping site, sometimes only transportation costs need to be paid. These sales and transportation costs can depend on the site to which the goods are shipped, therefore the different sites might pay different sales prices and transportation rates.

5.

BAAN IVc

5.1

Product Definition

Product data is always specific to a logistic company. This includes the bill of material, routing and the cost price. A product can have alternative routings and/or components, however, one product structure will be leading and will be the base for the cost price calculation. The calculation of a cost or fixed transfer price is done on a logistic company level and not on warehouse or work center level.

If a product can be produced in different ways, with different costs involved, multiple logistic companies have to be defined. Similar product data can then be synchronized with a data management tool.

Generic data, engineering data and product revisions are also stored against one specific logistic company. This means that e.g. engineering procedures will always only apply in the logistic company that is involved. Synchronization of revisions across the various sites would need to be controlled by a data management tool.

Customized or customer-specific products are always stored against a project code. This project code is attached to one logistic company. Once certain parts of the project are assembled in other production sites, inter-company purchase orders can be initiated. However, before this interplant purchase order can be received as an incoming sales order in the receiving site, the structure and contents of the project needs to be in place in the affiliated production sites. This can be done with a data management tool.

5.2

Planning and Forecasting

The planning of products can be executed on different levels. The lowest planning level is the warehouse level. By warehouse and product either replenishment orders, from another warehouse, or purchase or production orders can be generated.

The second planning level is on a logistic company level. In this case the planning will be executed on the totals of all warehouses in one company. However, once some inventory transactions need to be excluded from planning, it is possible to define non-nettable warehouses. Transactions stored against these warehouses will not be included in the overall planning run.

The highest planning level is on an enterprise level, covering multiple logistic companies.

5.3 Manufacturing Planning

Several planning methods are available for manufacturing planning. The lower level products in the product structure are often planned or replenished based on historical demand pattern either on a warehouse as well as on a company level. This is done with the SIC planning method.

Intermediate products in the product structure, those for which the demand is derived from parent items, can be controlled by a net-change, continuous or regenerative MRP planning. This planning is by product, product level and company. It is not meant to be used on warehouse or multi-company level.

Planning for the main products will be done using the Master Production Schedule (MPS). Within this master schedule, forecast and actuals will determine the planned receipts. The MPS can be defined on product level as well as on product family level (99 levels), information can be aggregated and exploded between products and families.

Based on a local master schedule, local receipts can be scheduled, either production or purchase orders. However, once one product can be produced in more than one company, the system will allow you to generate interplant receipts from one site into other sites. Priority settings, transportation times and available-to-promise can be used by the system to determine from WHERE the interplant receipts must be scheduled. This interplant scheduling is known as decentralized multi-site planning.

It is also possible to aggregate local data into a centralized company. The overall forecast can then be reviewed and reassigned to the local production sites. This is the centralized multi-site concept.

Capacity planning is always reviewed on company level, however with the proper naming conventions, multi-company views can be established.

The planning of customer specific products is also done on logistic company level by project. Once certain project parts are manufactured or assembled in other sites, purchase orders are used to trigger the demand in these sites. The project structure can be synchronized across the several sites with the help of data management tools.

5.4 Distribution Planning

The planning of stock transfers between warehouses is triggered by either local forecasts, actual orders or derived demand. Based on a bill of replenishment by item, item group or warehouse, the system will generate replenishment orders.

The Distribution Requirements Planning (DRP) logic only applies for warehouses within one logistic company. Inter-company stock transfers can only be generated by one of the manufacturing planning methods (SIC, MRP, MPS or PRP). The execution of this transfer goes via an inter-company purchase order. The system does support related inter-company EDI messages.

Since all prices and costs for a specific product within one logistic company are the same, there will be no invoicing between warehouses within a company. However, each warehouse has a finance company designation. Transfer of inventory between warehouses with different finance companies will automatically create intercompany transactions.

5.5 Subcontracting

In the routing it is defined which operation steps need to be executed in order to complete a production order. If one of these operations is done outside the manufacturing site, a link will be established between this operation and a subcontracting task, which is related to a subcontractor (either another site in the enterprise or an external supplier). Once the production order is released to the shop floor, a purchase order will be released to the subcontractor. Receipts against this purchase order reflect the progress of this subcontracted operation.

Sometimes, unexpected capacity issues will arise. In this situation subcontracting tasks have to be linked to the production order manually. Again purchase orders are used to trigger the demand and to control the progress of the operation.

5.6 Work Centers and Sub Work Centers

A physical production site can be identified with one or more shop floors or work centers. Linked to these work centers are the product routing tasks. This enables the shop floor manager or planner to view the planned and actual shop floor load.

When a company consists of multiple shop floors, it is possible to create an overall company view of available capacity and actual loads. This can be done by using sub and main work centers. Actual tasks are always attached and released to a sub work center, however, the system will automatically aggregate planning data into the parent or main work center.

This concept has been developed for a single logistic company environment, however with the proper naming conventions and table sharing, implementation in a multi-logistic company environment is possible.

5.7 Enterprise Information System

Within the Enterprise Information System (EIS) summarized management views can be retrieved from the available logistic and financial data. This information system will look like a multi-level ishikawa diagram consisting of many performance indicators (PI's). These indicators are fed from the application and can be recalculated using free definable formula's. The nodes of the diagram, attached to these performance indicators, can have different (warning) colours, depending on the predefined lower, average and upper limits of the PI's, compared to the actual application values of these PI's.

Within one ishikawa diagram, it is possible to represent multiple logistic companies. This is due to the fact that for each of the performance indicators a logistic company has to be defined from where the application values will be retrieved. This enables the user to get insight in the company business just in one diagram view.

6. BAAN IVc

6.1 Customers and Suppliers

In a multicompany structure both the static (name, address) and dynamic (open balances) data of the customer and supplier files can be centrally managed. The check on the customer credit limit against open balances can be done using centralized data.

6.2 Logistic and Financial Tables

Master data, such as countries, currencies, lines of business, can be centrally managed. This centralized data can be used by any company, regardless of the group company it belongs to.

7.

BAAN IVc

This chapter deals with general subjects that are related to data sharing and the data sharing methods. First definitions are given for some of the most used terms in this document. This includes the referential integrity of data, the type of network used and the client/server configuration. The last paragraph of this chapter introduces the case that is used throughout this part of the document to illustrate the data sharing methods.

7.1

Terms and definitions

Confusion easily arises when terms are not used consistently and in various contexts. This chapter explains how the terms data sharing, data replication, exchange, table linking and distributed transactions are used in this document and how they relate.

7.1.1

Data Sharing

Sharing can mean several things: to use with others, to possess in common, or to distribute in portions. In this document the term data sharing is used when two or more companies use of the same data records. Thus data sharing is used as a generic term that relates to WHAT is shared. HOW this data is shared, data sharing methods, can be split out in two types. These are discussed next.

Refer to Chapter 8 for further discussion on sharing data.

7.1.2

Data Replication

In this situation every company has exactly the same data, but each company has its own copy of the data . The same piece of data exists in multiple (physical) places; multiple instances of the same data. The process of copying the data is called 'replication' of data.

Data replication can be done by database table, by record and by column. In the first case the whole table is shared. In the second and the third case portions of a table are replicated between companies or databases. This situation assumes that the update of data is controlled to ensure appropriate consistency. Because reasons for sharing data vary, two examples are added to illustrated this:

Example 1: tiitm001 - Items. An item is manufactured in one company and sold to an assembly company. The item number, as well as specific data elements for the item can be the same, but in the manufacturing company the item type is "manufactured", in the assembly company the item type is "purchased". In this case the item can be entered in the manufacturing company, then portions of the item data can be replicated to the assembly company.
--

Example 2: tiitm001 - Items. In this example the purchase contracts for all items are centrally controlled in one company, but other companies purchase the items. In this case, all data for purchased items is entered in the contracts company, then replicated to the company that purchases it. Each purchasing company does not need to have the entire item file, only the portion that it purchases, while the contracts company has all purchased items.

7.1.3 Logical Table Linking & Distributed Transactions

If the same physical data is used by both companies only one table exists. Every piece of occurs only once; one instance of the same data. When the terms 'logical table linking' and 'distributed transactions' are used, one single physical instance of the data is used by users from multiple companies. When the company tables are on the same server, this can be accomplished by logical table linking. When multiple servers are used a kind of distributed transaction processing must be considered. Access to specific data can be restricted if required.

7.1.4 Exchange

Data exchange is a common term used in multisite situations. Usually it refers to data sharing or data replication or both. To avoid confusion with the Baan Exchange module the term data exchange is avoided. In this document the term Exchange is only used as reference to the Baan Exchange module. The Baan Exchange module offers functionality for data replications and is thereby a data sharing method.

7.2 Table References

Many tables include fields (data elements) from other tables. These fields are called referenced data. When data is shared, all referenced data can also have to be shared. This can depend on whether the data is mandatory or non-mandatory. Referenced tables are important in the referential integrity of data.

Where tables are logically linked or transactions are distributed, the referenced table will also need to be linked as well, especially if of mandatory data. If of non-mandatory data, fields can have different values in different companies as long as these values exist in the company referencing them. Thus the main table can be shared by using distributed transactions. The referenced table can be company specific or can be replicated.

For example: tiitm001 - Items is shared by more than one company. Many referenced fields exist in the item file, i.e. item groups, product types, etc.

If the item file is logically linked, the exact same values for all mandatory referenced fields must exist in both companies (i.e. item groups, etc.). These referenced tables are usually logically linked as well. If non-mandatory fields are not used (i.e. product type), the referenced tables do not need to be logically linked.

If a table is replicated, referenced fields can have different values, as long as these values exist in the company referencing them. In the case of non-mandatory fields, it is not necessary to have the referenced data shared if the field in the table is not used.

For example: tiitm001 - If the item file is replicated, programs must be able to reference all mandatory fields (i.e. item groups) within the company that is using the data. The value of the mandatory fields does not have to be the same in both companies, as long as the referenced value exists in the company referencing it. If non-mandatory fields are not used (i.e. product types), the referenced tables do not need to be replicated.

In some cases many levels of references will exist. The references can be far reaching and complex. For example the item file references such tables as item groups and other MCS tables, as well as engineering data, bills of material, routings and cost accounting data. For these reasons, and because tables can be shared in different ways in each implementation, table references must be investigated in detail for each implementation.

7.3

Networks

Networks and Client/Server scenarios influence the possibilities for data sharing. The performance of data transaction is different for both types of network. Thus the type of network has a great influence on the usability of most of the data sharing methods. Secondly some of the data sharing methods can or can not be possible depending on the client/server scenario that was chosen.

In each scenario a distinction can be made between the application (including B-shell), the database driver and the database. These three can be connected by means of a network. Two main types can be distinguished: Local Area Network (LAN) and Wide Area Network (WAN).

A LAN is a network of data lines within a physical location. The speed of data transactions is very good. A WAN is made up in most cases of rented cables that run over long distances. The capacity and the length of the cable restrict the volume and speed of data transfer. A WAN is generally more sensitive to breakdowns by e.g. physical damage.

In general when a WAN network is used, Baan does not recommend to do synchronous transaction processing. When it concerns asynchronous replication of data both types of network can be used.

7.4

Client/Server Scenarios

The distinction between application server, database driver server and database server, leads to three scenarios for a client server set up.

No Client/Server

When this scenario all companies are situated on the same database and all users have direct access to the database. Data sharing between companies can be accomplished by table linking. Users can login to the application server by means of local area or wide area network, whereby the WAN is not recommended because of low performance of on-line data processing.

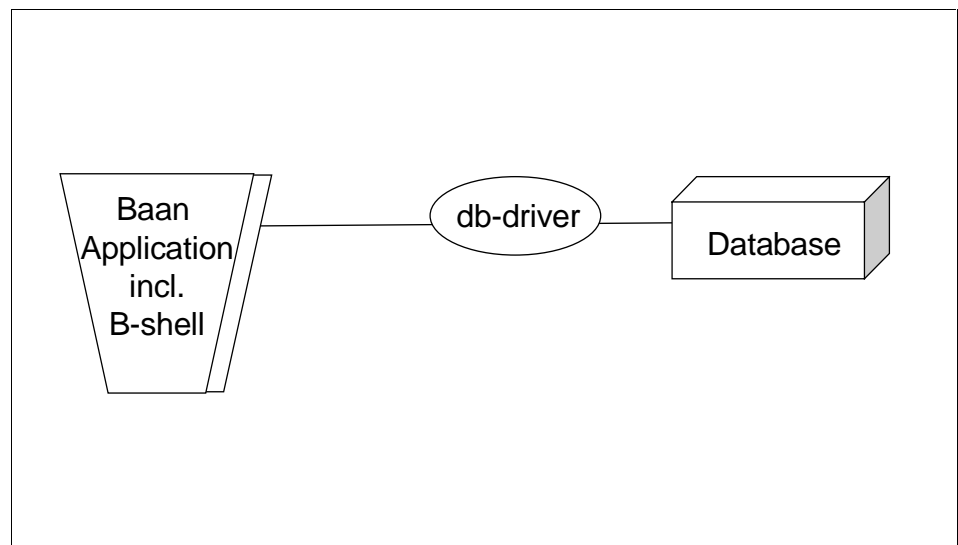


Figure 17 No client/server

Heavy Client / Light Server

In this scenario the database server is separated from the application server. The database driver can have links to multiple databases.

In principle it is possible to distribute the total set of tables of one company over multiple databases. In that case the user must have on-line access to both databases. This can be accomplished by the B-shell networking feasibilities, the Baan Distributed Database Driver or database triggers. However this can only be recommended when a LAN network is used. The single point of failure is very important. In the case one of the databases is down or the network is down, it is not possible to do any transactions anymore.

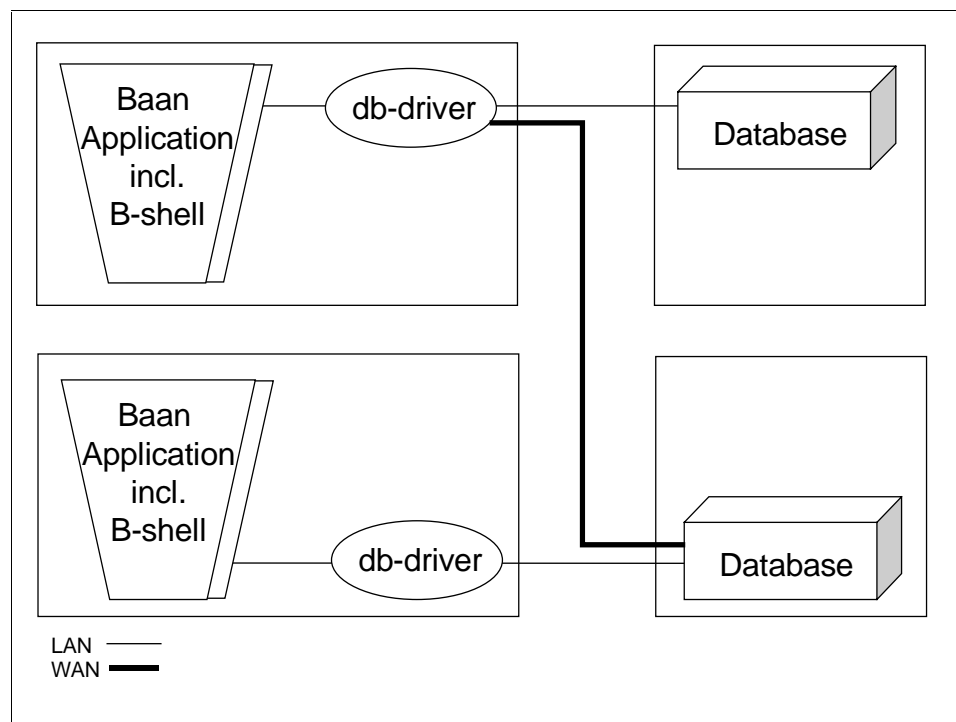


Figure 18 Heavy client / light server

When multiple companies are used, each company has its own set of tables. These tables can be placed upon separate databases. Users can need access to both databases for two reasons. In most of the multi-site scenarios companies must share certain types of data e.g. customer and supplier tables. Secondly the users could want to do transactions in several companies e.g. checking the quantity available-to-promise for a certain item in several companies. All methods for asynchronous replication and synchronous transaction processing are possibly usable over a LAN. Over a WAN only replication of data can be advised. Read the descriptions of the methods for characteristics, features and possibilities of each method.

Light Client / Heavy Server

The application servers are separated from the database and the database driver servers. The same comments as for the 'heavy client / light server' scenario apply. The Baan's Distributed Database Driver can not be used in this scenario.

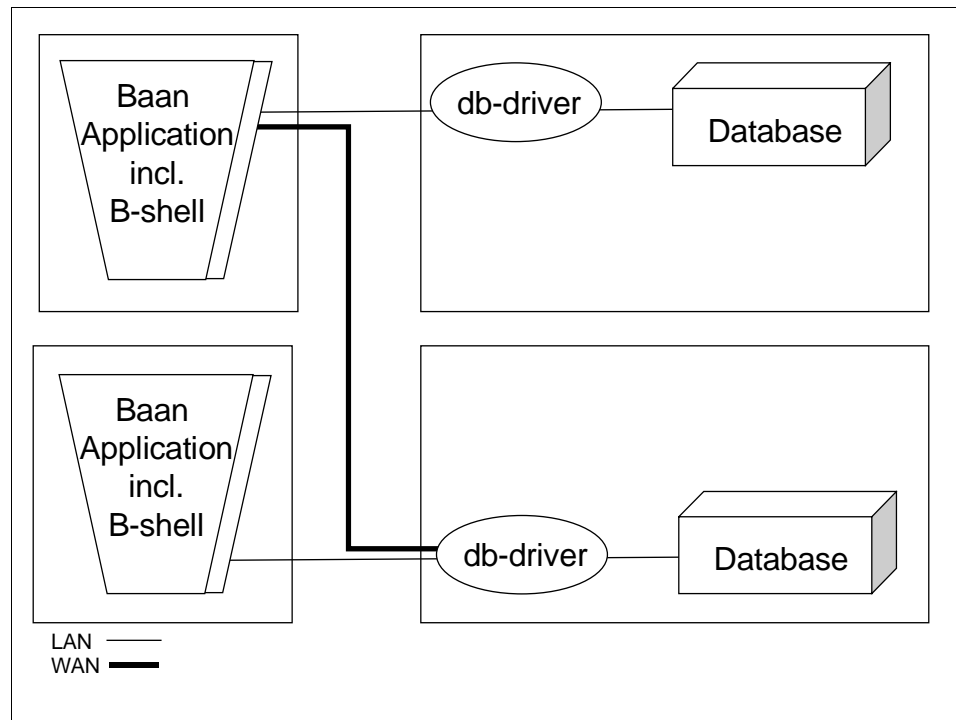


Figure 19 Light client / heavy server

Note

This document will not deal with the performance of the different brands of databases and database drivers. The Baan Its contains expected performance information for the different databases.

7.5

Choosing a data sharing method

Many factors play a role in the choice for the best method of data sharing. These factors can arise from the data sharing method. Each method has its advantages and drawbacks depending on the value of these factors. Further more each organizations has its own specific requirements, its specific organisation structure and its own specific transaction pattern. This makes every implementation unique and thus making it impossible to recommend a data sharing method from this document only. Implementation specific issues and requirements must be investigated.

Most of the influencing factors are discussed in detail in the functional or technical (data sharing) part of the document. They are listed here as a summary for the reader.

Technical factors in choosing a replication method:

- Number of servers
- Network reliability
- Network response time
- LAN/WAN Network
- Database capabilities
- Data model resemblance across servers
- Importance of data consistency
- Importance of data synchronicity
- Importance of working with actual data versus replicated data
- Referential integrity of data
- Total response time
- Read only / write actions
- Volume of the shared data
- Frequency of record changes
- Ownership of data
- What needs replication: data and/or procedures.

7.6

Introducing the case 'Remote Control'

Throughout the next chapters of this document a single example is used. The case illustrates the concepts and methods that are explained in the text.

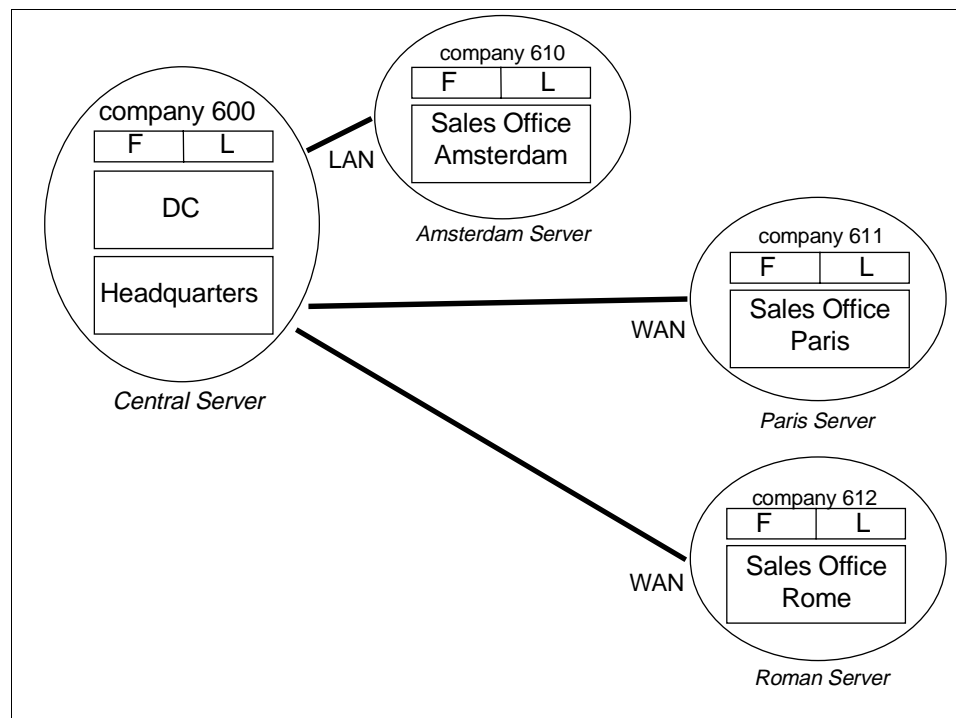


Figure 20 Remote control, multisite scenario

Remote Control

An American company selling consumer electronics has its European head quarters in Amsterdam. This consists of an European distribution center and the headquarters offices. Spread out over Europe 13 sales offices are in place. The Dutch sales office is also located in Amsterdam. The multisite structure is set up as a multi logistic / multi financial scenario. The head quarters and the distribution center form logistic company 600, which is also the group company. All sales offices are separate logistic and financial companies.

Sales offices are responsible for sales and marketing in their area. Every sales office is owner of the customer records for the customers in its own area. The head quarters office is responsible for the financial reporting. Head quarters owns a/o the static financial data (tax codes, ledger codes) and the item tables. The central distribution center in Amsterdam performs inventory and distribution planning and is responsible for the planning, order and inventory data.

The head quarter offices and the distribution center work on the central server. The Dutch sales office has a local server that is connected by a local area network to the central server. The other European sales offices work on separate servers and are connected to the central server by a wide area network.

8.

BAAN IVc

In some cases sharing data is optional, and is done for reasons such as resources, security and data consistency. Sharing is preferred to manually updating several companies with the same data. In some cases sharing data is mandatory, certain functionality requires sharing specific data.

Types of data commonly shared are static data, which is usually master data and includes customers, suppliers, ledger master, pricing and contract data. Dynamic data, which is usually transaction data such as purchase orders can also be shared.

Because every organization has unique data management requirements and hardware configurations, the methods of sharing data, and the decisions of which data to share will be unique for each implementation.

8.1

Multicompany Structures

In any group company structure it is mandatory to share certain tables. For a detailed discussion about sharing tables in specific multicompany structures, refer to appendix A. The requirements support the group company functionality. Unless otherwise noted, the tables listed must be shared.

Multicompany environments usually involve a group company structure. However, this is not always the case. Data can be shared across group companies and between stand alone companies (single logistic / single finance), as well as within a group company. In these cases sharing data is not mandatory. Refer to appendix B for information on the functional aspects of sharing data.

If data sharing is required across group companies or stand alone companies for any table with a **Finance Company** field (i.e. warehouse, work center, projects, customers, suppliers, first free numbers, purchase, sales and production orders), the tables must not be logically linked. Caution is advised while building replication schemes to properly determine the value for the Finance Company field.

Refer to appendix G for information on setting up a company structure.

8.2 Data Sharing by Package

Beyond the mandatory shared data for a particular structure, see above, functional groups of data can also be shared as listed in appendix B. However, care must be taken to thoroughly analyze the business requirements and data sharing considerations. The information in the appendix is not intended to be definitive, but a guideline to be considered along with business requirements and technical architecture. These guidelines are just that – guidelines. Table references and implementation specific uses and issues must be considered.

The tables are presented in functional groups. The data in each group can be considered independently, however the use of a specific group on its own can result in dependencies on other data. It is not always mandatory to share all tables in a group. Each table must be analyzed to determine if dependencies on other tables requires further action.

The appendix only refers to the most commonly shared data in the Manufacturing, Distribution, Finance and Common packages. Other data within these packages, as well as data in other packages, are not mentioned in this document, but can also be shared.

8.3 Multiple Data Servers

In the case of multiple data servers, Baan does not recommend logically linking tables across data servers. However, group company functionality requires that some tables are shared. Baan IVb introduced the primary company designation which provides group company functionality without requiring tables to be logically linked across data servers. One company on each data server other than the server on which the group company resides, must be the primary company for the server.

In all group company structures, the tables that are normally logically linked to the group company are logically linked to the primary company. On the group company's server, the tables are logically linked to the group company. Replication of data between the group company data server and all other data servers is required.

Baan IV also provides for central customer and supplier tables, without logically linking these tables. This supports central order and invoice balance data, including credit checking. Replication of data between all instances of these tables is required.

9.

BAAN IVc

9.1

Introduction

The Baan application offers two types of replication: the Full Exchange and the Audit Based Exchange. All database systems offer functionality for replication data across servers. In the market some independent replication packages are available. Each of the methods is discussed by a description of functionality and the possibilities and limitations. Each paragraph is concluded by a summary.

Functional Description

For the asynchronous methods the local servers need to have a copy of the data records of the central sites. When a transaction is done the local records are updated. On running the replication the remote records are updated. The records in both servers is not updated at the same moment. The more frequent the replications are done the more accurate data is available.

Remote control

For the illustration of asynchronous replication methods the replication of static and financial data is discussed.

The head quarters in Amsterdam is responsible for financial reporting of all European offices and warehouses. In every multisite situation certain static data must be shared e.g. customer and supplier tables, tax codes, item data, warehouse codes, etc..

In this situation the tax codes and item codes are owned by the headquarters. They do not change very often. It has been decided that tax codes are replicated to the remote servers once a week. Item data is replicated on a daily basis.

The sales offices own the customer files of the customers in their region. Every night the customer files are replicated to the central server.

Possibilities and Limitations

The following comments are valid for all asynchronous replication methods. With asynchronous solutions a two-phase commit can never be done. Compared to synchronous methods the asynchronous method often offer a better response time to the user.

The existence of multiple instances of the data records requires a good control over data ownership and replication schemes. In this a master-slave relation is recommend. Ownership can de arranged by using first free number functionality and setting authorisations.

Replication can be done for different levels of detail; replicate tables, replicate a set of records and replicate a specified set of rows and columns. The most detailed replication schemes require more maintenance. Together with the replication schemes the ownership of data must be arranged.

Summary

Characteristics of Asynchronous Data Replication methods	
Multiple instances of the same data records.	Data equality is dependent on updating and replicating frequency.
Transactions are always done on local records.	Ownership of data is best arranged by a master-slave relation between origin records and replicated records.
Usually gives a good response time to the user.	Replicate by table rather than by records or row/column

These characteristics apply to all asynchronous replication methods that are discussed in this document.

9.2

Exchange

Functional Description

Exchange is the Baan solution for the import and export of data between applications. The Baan Exchange module handles two types of replication; Audit Based Exchange and Full Exchange of records. The features described here apply to both the Full Exchange and the Audit Based Exchange unless stated differently.

Exchange is part of the Baan application and is database independent and network independent. On import or export of data the business rules in the Baan Data Dictionary are checked.

<p>Remote Control</p> <p>The Paris office is the owner of the customer records of all French customers. Every week the customer records are replicated by a Full Exchange to the central server in Amsterdam. Business wise ownership of the customer records is arranged by the ranges and authorizations; all French customers numbers start with a four. By means of export schedules during export only those customer records are selected that start with a four.</p>

The Exchange can be manually executed or scheduled in a Baan job. It can run at a predefined schedule or according to a calendar. The Exchange schemes must be set up in every site that wants to import data records from other servers or exports data records to other servers.

Possibilities and Limitations

In Exchange export schemes can be build to export tables, records or row/column combinations. Replication of whole tables is recommended. On the remote site the import scheme can be built in such a way that specified rows and columns can be excluded from import.

Import and export schemes have functionality to transform data before it is imported or exported. This includes the following: values can be raised by a factor before insertion, constants can be assigned to certain columns and calculations can be done. On import of data default values can be given to specified columns.

Import schemes have functionality to control import by type of action i.e. insert, delete, modify and update. This enables sites to possibly enter new records, but not change existing records or visa versa. Import schemes can be build to start functions based upon the import of data. This requires the use of scripts.

The data updating in the Exchange module is uni-directional. Ownership of data must be clear in order to keep track of changes to the data. Functional ownership of data can be arranged even to the 'field' and 'row' level by means of the authorizations in the Tools package. However this can require a lot of maintenance. In setting up proper Exchange schemes the 'technical' ownership of data (master-slave relation) must be secured by using proper import and export schemes.

The Exchange module can not guarantee arrival of the exported records at the remote server or import of records into the remote database.

When using Exchange the data models on the servers need not be exactly the same. The Exchange module will check the data integrity; a sales order line must always have a reference to a sales order header. If this is not the case, the records will not be inserted on the remote site. After repair the refused records can be imported by a restart from the log-files.

The Exchange module can export data to non-Baan applications also. In that case some extra effort must be made for the programming of checking and programming rules.

Summary

General Characteristics of Exchange	
Database and network independent	Import can be controlled by type of action: insert, delete, modify or update.
Manually executed or scheduled by a Baan job.	Data integrity is checked before import.
No centralised replication control.	Ownership of data must be arranged.
Import and export schemes can be build to include and exclude tables, records or row/column combinations.	No guaranteed replication.
Exchange makes use of the business rules in the Data Dictionary.	Data models need not be the same.
Transformation of data upon export on import.	Exchange can export data to non-Baan applications.

See also the characteristics of the Full Exchange and the Audit Based Exchange.

9.2.2

Full Exchange

Functional Description

The full exchange is based upon the data tables. A selected range of data is exported to ASCII files. These ASCII files can be copied to and imported into remotes databases. The full exchange will export all selected records (rows and/or columns) regardless the antiquity of the data.

The full exchange is mostly used in less frequent updates e.g. the export of data to history files or the updating of employee files. The export is manually executed or is scheduled by a job.

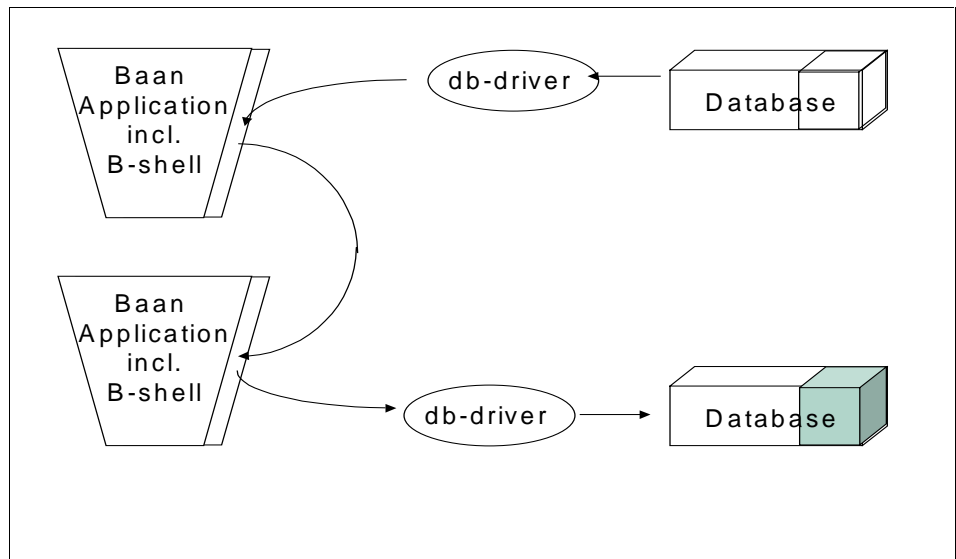


Figure 21 Full exchange

The Full Exchange of data is table number oriented. This means that the order in which the tables are replicated can be different from the order in which the application would update the records.

Remote Control

Item data is replicated to the remote servers every night by a full exchange. During replication the item type 'manufactured' is replaced by 'purchased'. The order lead time is replaced by purchase lead that represents the transport time from Amsterdam to the sales office.

If the sales offices would alter item data, the changes would be lost on the import item data from the central server.

Possibilities and Limitations

All features described for the Exchange module in general apply.

Only the full exchange can handle the replication of texts.

Summary

Characteristics of Full Exchange	
Data records are converted to ASCII files	Most feasible for less frequent replications
Can handle replication of texts	

9.2.3

Audit Based Exchange**Functional Description**

The audit based exchange is based upon the contents of the audit log records. All transactions within a specified range of time are filtered out and are exported to ASCII records that can include inserts, deletes and updates. These records are copied to remote sites.

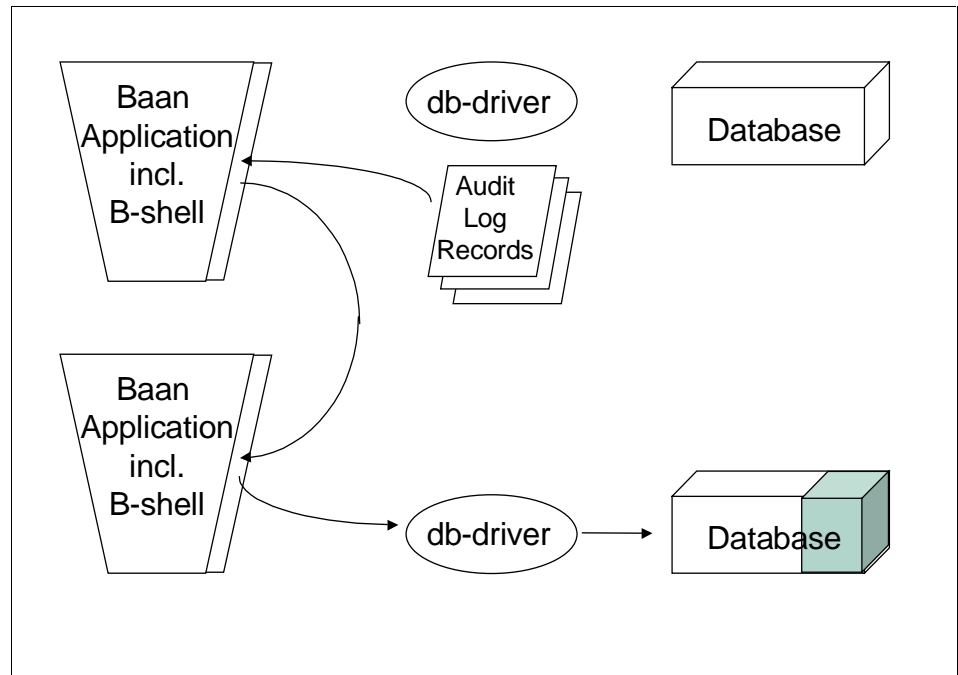


Figure 22 Audit based exchange

This type of replication can be triggered by Baan jobs, which collect all transactions within a certain range of time. The audit based exchange is mainly used for the batch replication of data. Only the records that have been changed will be replicated. The minimum interval of replication is in most situations 15 minutes. If a lot of records need to be updated performance decreases. Therefore the audit based Exchange might not be suitable for very dynamic data.

Possibilities and Limitations

In the import/export schemes a selection of rows and columns can be made. In the Audit Based Exchange schemes the same functionality is available as described with exception of the transformation of imported data values. This is not possible for the Audit Based Exchange.

The fact that the replicated records originate from the audit records can lead to in-doubt transactions. A in-doubt transaction can result from the fact that the logging in de audit log records might be different from the loggings in the database records due to network or server breakdown. See the Appendices for more information about in-doubt transactions.

The Audit Based Exchange can not be used for the replication of texts.

Audit Based Exchange is table oriented on the exporting site. Most tables will have their own log records. During export the log tables are exported in the table number sequence. Upon import the imported records are merged according to the transaction sequence, as recorded by the date and time in the log records. However in some cases the algorithm is not always reliable. The transaction sequence can not be restored in some cases.

Summary

Characteristics of Audit Based Exchange	
Based upon the audit log records	Possibility of in-doubt transactions
Contains only the changed values and records	Not feasible for text replication
Minimum replication interval is 15 minutes. Not possible to do transformations of imported values	Import of records is transaction sequenced, however the algorithm is not perfect

9.3

Database generated replication

Functional Description

This type of data updating is prompted by the databases. The major commercial vendors are Oracle, Informix, Sybase, SQL Server (Microsoft) and DB2/6000 (IBM). All named database systems offer functionality to update data on remote servers. This functionality differs substantially from the functionality the Exchange module offers. This paragraph describes the main differences with the Exchanges module functionality. Detailed information about the respective database systems, can be acquired via the vendors of these systems.

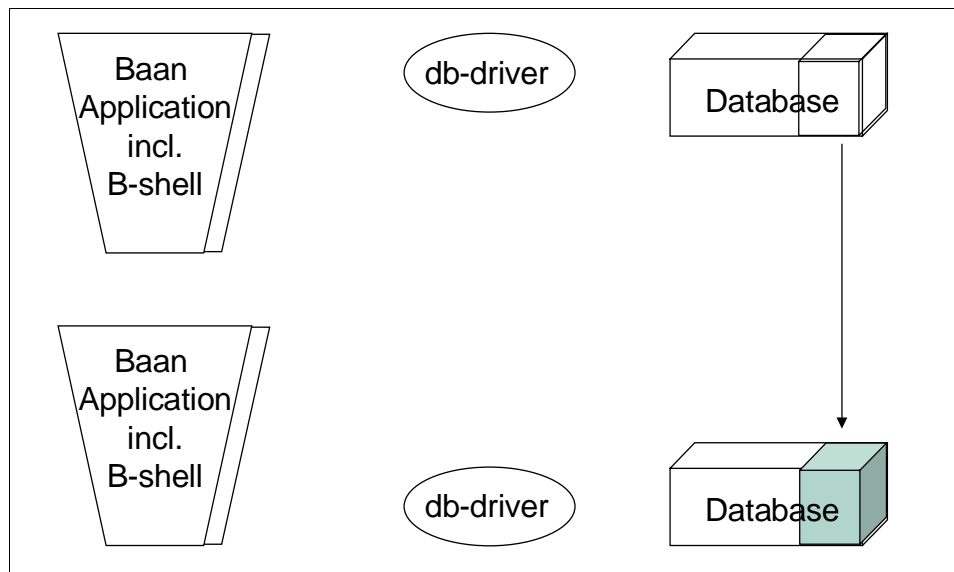


Figure 23 Database generated replication

Remote Control

The central distribution center in Amsterdam needs near actual information about financial transactions. Because of the high volumes a database generated replication is chosen. Every two hours the financial transaction data of the sales companies is replicated to the central server.

Possibilities and Limitations

The database generated replications can be more frequent or even continuous resulting in a near on-line replication. Usually the performance is better and larger volumes can be handled. The export of data can be event driven, meaning that whenever the data in the local records is updated, directly this will be replicated to the remote server. In this case the sequence of the transactions will be followed and replications occurs continuously.

Deliveries are guaranteed as long as the network is not physically damaged. Before the transfer of the records over the network, the records are kept in a queue. If the network goes down the records are stored in the queue and will be transferred as soon as the network is up again.

The main disadvantages lay in the fact that these type of exports can only handle the export and import of data without any reference to the business rules in the Baan Data Dictionary. This results the disadvantages discussed below. Database generated replication methods will not maintain referential integrity of the data. For instance a sales order line must always be linked to a sales order header. Upon database generated updates this integrity can not be checked. The data models of the Baan applications must be nearly the same on both servers. Whenever the Baan Data Dictionaries differ substantially data conversions will go not be possible.

Texts can be replicated, but one must take into consideration that text numbers are not necessarily unique across database systems. Database generated replication methods can no handle data value conversions.

When an update is made via the Exchange module (i.e. Baan application) the reference counters and hash columns are updated. This is not done by the database generated updates. This can result in inconsistencies within database records. In the Appendices reference counters and hash columns are explained in more detail.

Domain constraints are not checked either. A domain constraint can be the rule that a certain domain (e.g. sales order number) always starts with a certain number. Upon RBDMS replication this logic is not checked.

In replication triggers can be used for two functions. Firstly whenever procedures must be invoked on the remote site, triggers must be used. Secondly triggers can be used for the 'mirroring' of databases. As soon as a record is inserted, updated or deleted, trigger might be fired to do the same on a remote database. This results in an event driven, on-line replication. Trigger functionality is very database dependent and some database systems do not

offer trigger functionality. Trigger functionality is also described in the paragraph Synchronous Transaction Processing.

This type of replication is in general only possible when the sending and the receiving database are of the same vendor. Only Sybase offers some functionality to deal with data replication to or from a different database type.

Summary

Characteristics of Database generated replications	
Triggers can be used to enable referential checks and on-line replication	Data integrity is not checked
Functionality dependent on the type of database	No transformation of data during replication
Higher volumes and frequency possible	Text can be replicated under a few conditions
Event driven replication possible enabling transaction sequenced near on-line replication	Hash columns, reference counters and domain constraints are not replicated possibly resulting in inconsistencies between databases
Deliveries are guaranteed	Only Sybase allows replication of data between databases of different vendors

9.4 Independent data replication packages

Functional Description

Independent packages are available for data replication across databases. Information about these packages, can be acquired via the vendors of these packages.

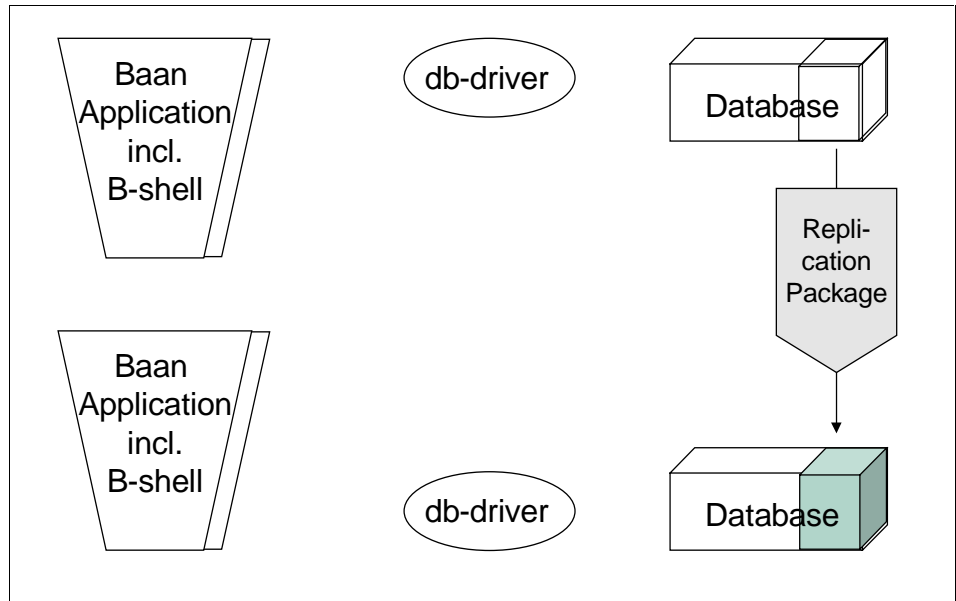


Figure 24 Independent data replication packages

Possibilities and Limitations

In general they are more easy to use, but have more limited functionality. In general the performance is better.

The main advantage is that these packages can handle data replication between different types of databases. However these packages cannot be used on the Triton Base database.

An example can be the Omni replicator by Praxis.

10.

BAAN IVc

10.1

Introduction

Functional Description

With synchronous transaction processing the user always is able to use the actual data. Of every data record only one instance exists. A single transaction can have effect on two or more records. These records are not necessarily stored in the same database and on the same server. In distributed transactions both records are updated simultaneously. Two-phase commit assures that either both of the record changes are executed or that none of the record changes is executed, thus assuring data consistency across servers.

Remote Control

When is chosen for distributed data transactions an on-line network connection must exist between the servers. The Dutch office is connected to the central server by a LAN. This makes it possible to have on-line contact with the central server.

The servers of the other sales offices are connected by a WAN. Because of the performance of the WAN data replication is used in stead of distributed transactions.

The transaction can be distributed on three levels; the b-shell, the database driver level and the database level. All three will be described, but not all three have a two-phase commit. Logical table linking is also included in this chapter. Table linking is not distributed transactions. Both in logical table linking and in distributed transactions the user accesses the actual data records: synchronous transaction processing. The difference is in the number of servers. For logical table linking the tables of all companies must be on the same server.

Possibilities and Limitations

Distributed transactions always use actual data for the updating of records. The update or query is performed real time. The advantages can be limited by the network response time and the network reliability. A query on a database will be send over the network. The user must wait for the remote server to respond. Especially on a Wide Area Network the response times can be long when many transactions need to be done. When lots of transactions are entered simultaneously a batch update often results in a better performance.

The major issue of concern is the 'single point of failure'. If the network or the database server goes down none of the application servers can work, because no one can get access to the shared tables.

Summary

Characteristics of Synchronous Transaction Processing	
One instance of every data record	Not all methods have two-phase commit functionality
The type of network used influences the response time greatly	The single point of failure must be considered
Synchronous Transaction Processing is dis- advised when using a WAN	

These characteristics apply to all synchronous transaction processing methods that are discussed in this document.

10.2 Logical Table Linking

Functional Description

In a multisite situation often the same data must be shared among Baan companies e.g. the use of the same suppliers records across different Baan companies. When all companies are on the same server the tables can be linked by means of logical table linking. A query or data input is redirected by the Baan application to a physical table of another company.

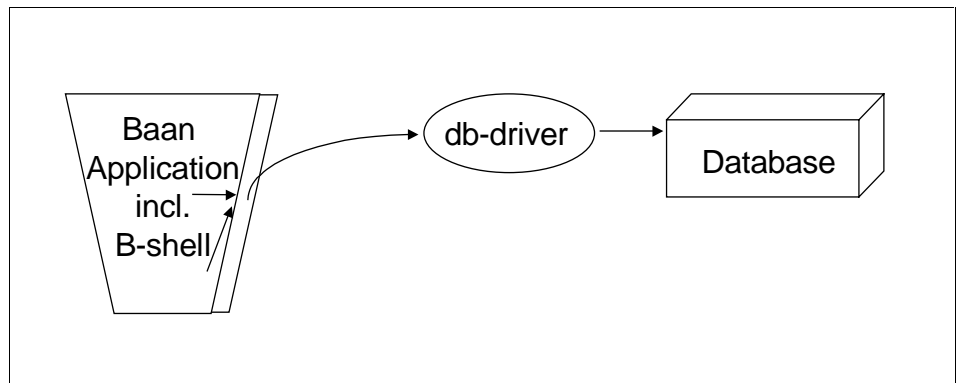


Figure 25 Logical table linking

Remote Control

The sales offices are separate logistic companies. Some tables of the Dutch sales office (company 610) are linked to the tables of company 600 (headquarters and distribution center).

When the Dutch sales office enters an order the system will use the local area network to retrieve item data and customer data from the central server.

Possibilities and Limitations

Table linking can only be used when the data models of both companies are exactly the same. In table linking the whole table is shared. It is not possible to link certain rows or columns of a table.

When main tables are linked, the referenced tables must also be linked. Tables can include fields (data elements) from other tables. These fields are called referenced data. When mandatory data is referenced and is shared, all referenced tables must be shared to, in order to keep consistency between tables. For non-mandatory data referenced tables need not be linked logically, but the values must exist in the company that references them.

The general comments about tables are also valid for logically linked tables. When tables are linked one must be aware of the expected size of the table. In deleting actions the table references need to be checked in all companies that have access to the table. The time this takes is depending on the size of the table and on the number of companies involved. Queries on big tables are more time consuming. A factor of lesser importance is the number of users of a table. For updating sessions records might be locked when two users are trying to access the same record. The chance of this happening grows with the amount of users.

Linked tables need not be on the same physical location. This is not a problem, because the B-shell will direct the data to the right database. However linking tables over a Wide Area Network is not recommended, because of performance reasons. See also Baan Network Features using the B-shell.

Summary

Characteristics of Logical Table Linking	
Data models need to be the same	Be aware of the number of users
Users share the whole table	Linking over a WAN is strongly dis-advised
Be aware of the expected size of the table	

10.3

Distributed transactions using the Baan B-shell

Functional Description

This is multi server functionality using the B-shell application. For the Baan application it is not important where data is stored. It will direct each updating, deleting or insertion request to the correct database.

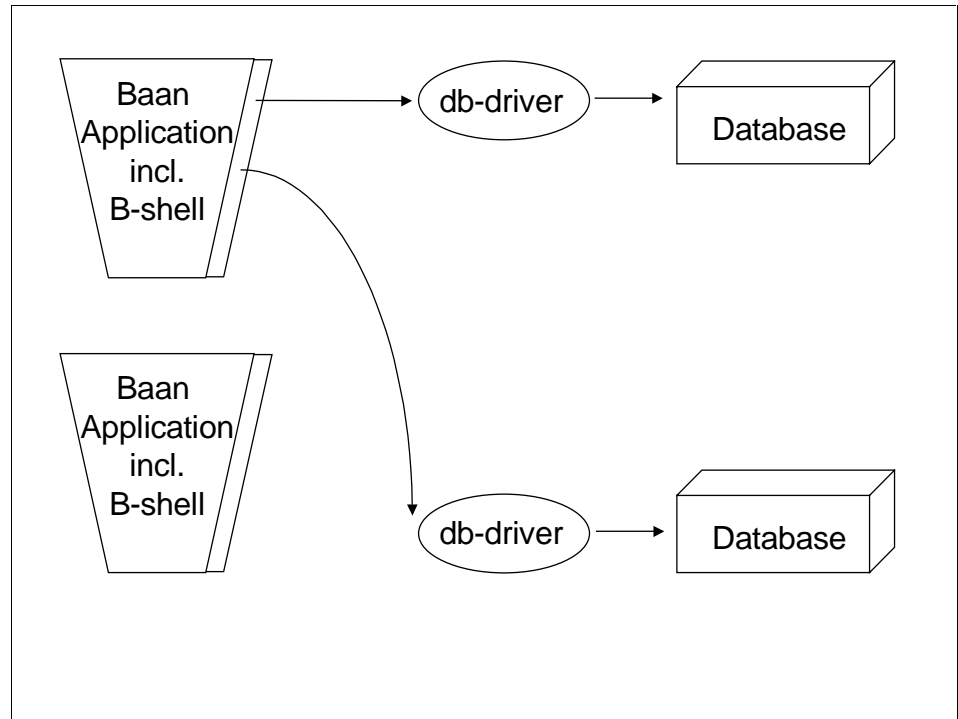


Figure 26 Distributed transactions using the Baan B-shell

Possibilities and Limitations

A difference can be made between updating and reading of data. If only read actions are done on the remote database, A WAN might be used. But aware of the response time and the single point of failure.

On updating of data the Baan B-shell is not able to check if records on both servers have been actually updated. The Baan B-shell is not able to perform a two-phase commit. If during the transaction any database or network problems occur, this will result in asynchronous data. Synchronising the tables again is difficult. Network reliability is very important in this case. Therefore this method is forbidden on WAN's.

It is possible to have the B-shell direct each update to two database drivers. This is called mirroring of data bases. This is not a distributed transaction, but a form of replication. The second database might serve as a backup or a database which is used for special purposes.

Summary

Characteristics of Baan Network Features using the B-shell	
Deliveries are not guaranteed	Forbidden on a WAN for updating actions. Maybe a LAN can be used for reading actions
No two-phase commit	

10.4

Distributed transactions by Baan’s Database Driver

Functional Description

Oracle offers two possibilities for the handling of distributed transactions. The use of Baan’s Distributed Database Driver and synonyms.

The Baan Distributed Database Driver is currently only available for the Oracle database (level two technology). The driver can on-line select and update local and remote databases within one transaction. The db-driver has one connection link the local database and has access links to other databases. The db-driver makes it possible to use Oracle’s two-phase commit protocol.

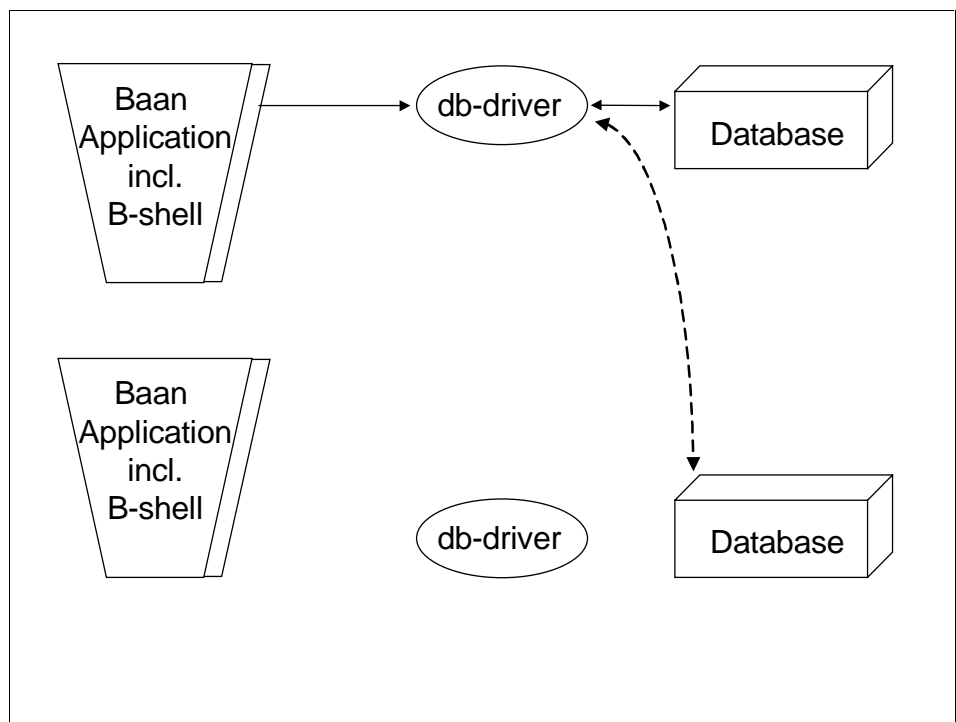


Figure 27 RDBMS solution by the database driver

A second possible method in Oracle is the use of synonyms. The b-shell will sent an instruction to the db-driver to update a specific record in a certain database. The db-driver substitutes the ‘address’ of the database and redirects the instruction to another server. The substitution is done by means of an synonym.

Remote Control

The company has decided that the Dutch sales office has on-line access to the planning and inventory data of the distribution center. On sales order entry the available to promise needs to be checked in the MPS planning. Via Baan's distributed database driver the ATP data is retrieved from the central server. When the order is confirmed the sales orders are kept on the remote server. Simultaneously with the confirmation of the sales order inventory must be allocated and financial transactions must be generated and stored on the central server. Oracle's two-phase commit will guarantee that those actions are performed on both servers.

Possibilities and Limitations

The two-phase commit protocol relies on the fact that both databases are Oracle. When databases of different vendors are used, two-phase commit is not possible. The functionality for select statements in the Oracle database is not very strong and this can cause performance problems. Besides that administration and maintenance of the systems is more complicated. When transactions need to be distributed over a lot of sites performance is lower and the risks are higher.

Synonyms create a lot of network traffic across the database link, as many single-table/single-row queries are send over the network. Creation and maintenance of synonyms is complicated and labour intensive. Therefor this method can not be fully recommended.

In both cases the Baan B-shell will assume that all data is stored in one database. The audit log records are created based upon the B-shell instructions. When above methods are used the audit log records do not contain the correct information. In the case the same record is updated from multiple db-drivers the different audit log records can contain confusing information. This can be worked around by making the log records incremental in which case the date and time of the transaction is logged. When the backup is rolled back to the database the records can be selected on date and time.

The use of a Local Area Network is preferred with respect to network response time and reliability.

Summary

Characteristics of Distributed transactions by Baan's Database Driver	
Baan Distributed Database Driver	Synonyms
Each db-driver has one connection link to a database and possibly has multiple access links to other databases	Create a lot of network traffic
Only available for Oracle level-2 technology databases	Creation and maintenance of synonyms of labour intensive and complicated
Two-phase commit possible if both databases are Oracle level technology	Use of synonyms is not advised
Select statement are not very strong	The audit log records contain incorrect information

10.5

Distributed transactions using triggers**Functional Description**

Distributed transactions can also be accomplished by means of trigger functionality in the database. Triggers are defined on a table field level in the database. The trigger is linked to a stored procedure. When a value is inserted, updated or deleted the trigger is fired and this will start the stored procedure. The procedure can be programmed to mirror databases, to perform distributed transactions with a two-phase commit and to check referential integrity.

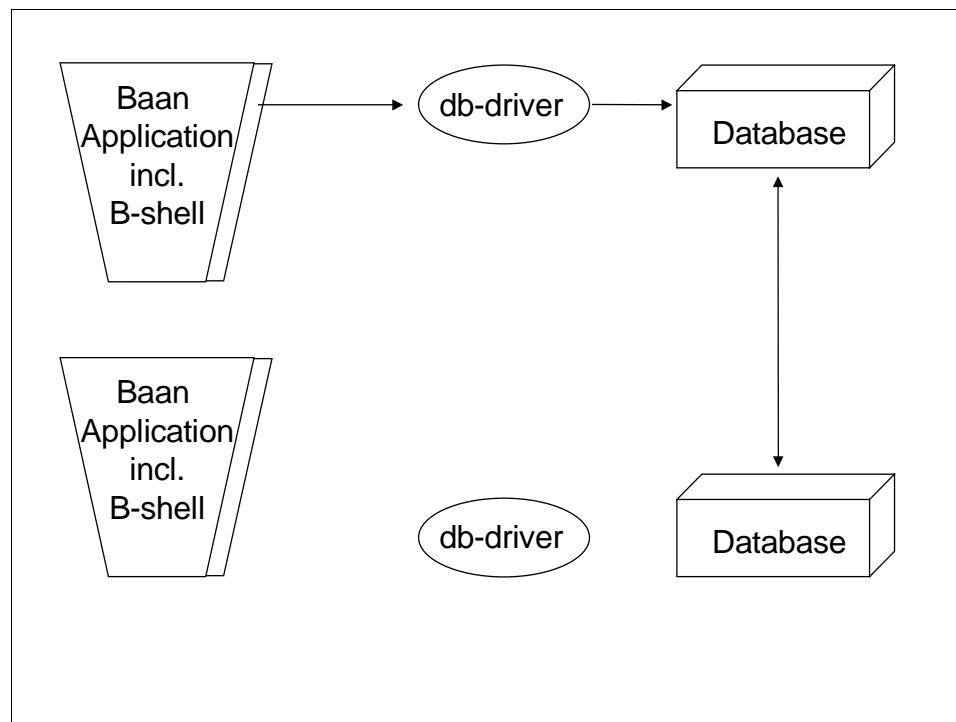


Figure 28 RDBMS solution using triggers

Possibilities and Limitations

The trigger functionality is not available for every RDBMS.

In distributed transactions the trigger will perform the same action on the remote database. If of replication the procedure the trigger will carry the instruction to insert the same value on one or more databases. The trigger procedures can be defined in a way that a two-phase commit will be performed. Upon an insert, delete or update action of a child record the procedure can also contain instructions to check upon the parent relation, by which referential integrity of data can be maintained.

Hash columns, reference counters and domain constraints can not be checked by trigger functionality. See also the Appendices.

The construction of triggers is a programming based upon user requirements. Triggers are defined manually in the database. Administration and maintenance is complicated and labour intensive. Therefore this method is not advised by Baan.

Only the insertions, updates or deletions are performed by the database triggers. The selection of records (select statements) have to be performed by the Distributed Driver.

Summary

Characteristics of Distributed transactions using triggers	
Triggers can be build to perform distributed database transactions, mirroring of tables, to perform referential checks of data and two-phase commit	Creation and maintenance of tables is complicated and labour intensive
Hash columns, reference counters and domain constraints can not be checked by triggers	Not advised
The select statements must be don by the database driver	

In a multisite environment probably not all sites are upgraded at exactly the same moment. This results in an intermediate situation in which multiple versions of the Baan software are used together. In such a situation conversion programs are needed between applications. These conversion programs must convert two-directional. It must be possible to transfer data to the higher version and it must be possible to be transfer data from the higher to the preceding version. Conversion programs to the higher version can be supplied by the Baan Company. Conversion programs for the reverse must be custom build.

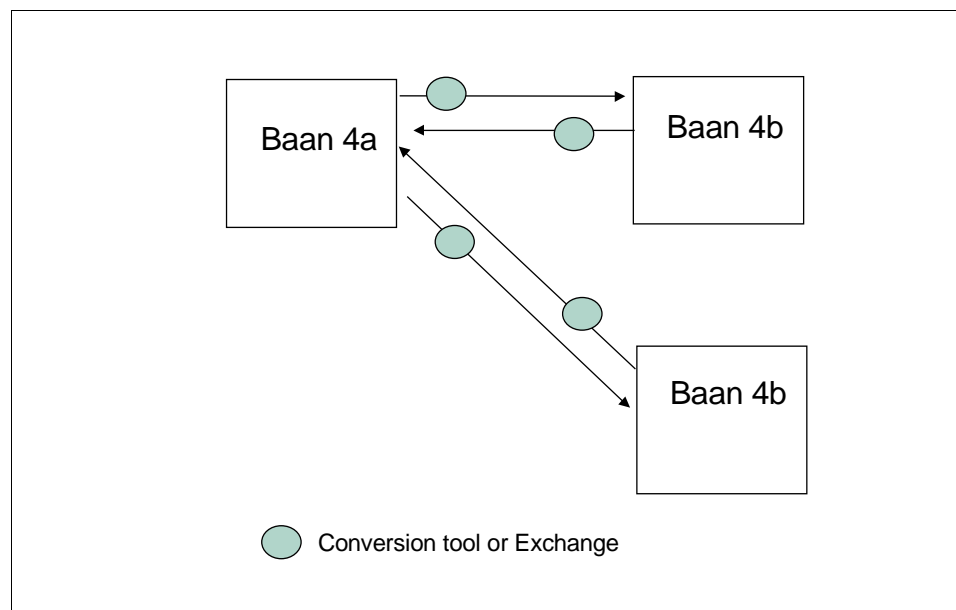


Figure 29 Multiple versions together

The first concerns is the nature of the key fields. When key fields are the same in both data models, the data records can possibly be shared.

When using Exchange the export and import schemes can be built in a very detailed level, facilitating data replication between different versions. Useful features of the Exchange module are its ability to perform data transformations and the check on referential integrity of data upon import of data records.

The replications that are generated by databases also have functionality to select rows and columns, thus facilitating the replication between different Baan data models, however the data models must be very much the same when using database generated replication methods. Database generated replication can not do data conversions and does not check data integrity. This is a possible source of data inconsistencies.

Table linking is not possible across companies that work with different data models. Table linking means that the whole table is shared; all columns and all rows.

Depending on the nature of the upgraded functionality it can be clear that in some cases the new functionality will only be available when all applications have been upgraded.

The Baan multisite MPS is capable of processing plan data from multiple companies when running the MPS. In a multisite situation not only multiple companies might be used, the companies can also be situated on separate servers. The Baan software is able to run a multisite MPS in such a situation, however performance is severely affected. This is a result of the way the multisite MPS run is executed.

A multisite MPS run can be divided into three parts.

- Firstly the Supply Chain structure and the Bill of Critical Material is checked. This is done centrally. The Supply Chain Structure, Plan Items and Bill of Critical Materials are used to construct a multisite Bill of Materials. This determines the order in which the plan items are processed. For every plan item the order in which the local MPS's must be run is established.
- Secondly a series of local MPS runs is executed by plan item. Each local MPS consists of three steps: retrieve data from the local company, process the data centrally, write the data into the local company. A multisite MPS includes many of these runs. The multisite plan co-ordinates this.
- Thirdly the multisite MPS run concludes with a run for the central plan in which local data is aggregated to the central plan. Exception messages are only generated for the local MPS's.

The second and third steps are repeated for every plan item.

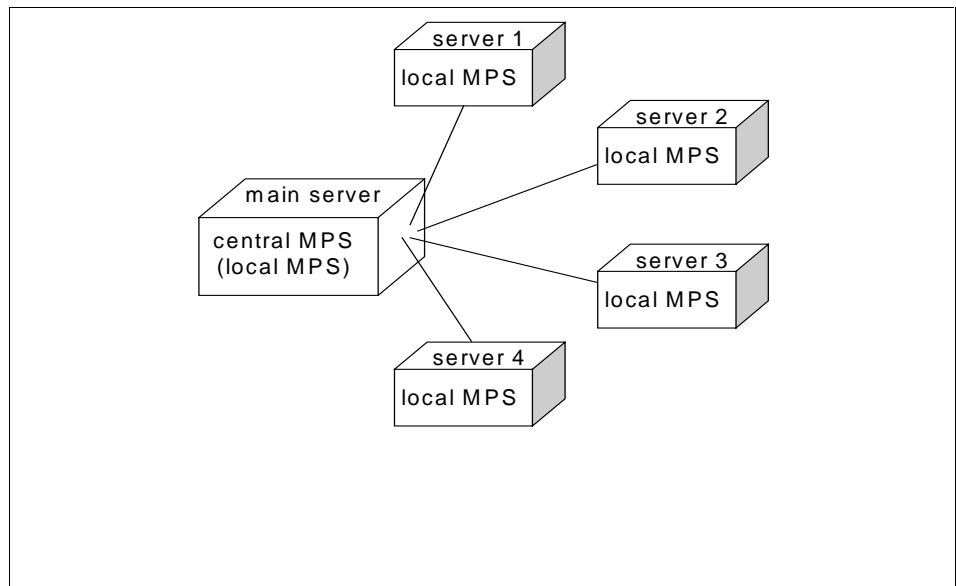


Figure 30 Multisite MPS on multiple servers

During the multisite run data is retrieved from and written to local companies many times. When the local MPS's are on separate servers this will lead to a lot of network traffic making the multisite MPS run very susceptible to network failures. The response time of the network has great influence on the performance of the MPS run.

A good alternative is the replication of the proper data to the main server only just before the execution of the central MPS run. A proper replication method must be chosen to do this. After the central MPS run the company data must be replicated to the local servers. Basically a multisite MPS on a single server environment is performed. Performance and reliability are much better. This procedure does not influence ATP and forecasting functionality.

In the Baan 4b software the performance of the multisite MPS has been improved by using multiple B-shells. Every B-shell will deal with one or more companies, thus enabling parallel processing. This has been designed for use on one server.

13. Appendix A Sharing data in multicompany structures

The information in this appendix is valid for BAAN version IVc4. See also 'Multicompany structures' in Chapter 8 for an explanation of the information in this appendix.

13.1 Multicompany Environments

Multicompany environments usually, but not always, involve a group company structure. In order to communicate between companies (i.e. EDI, MPS), regardless of the structure, the following table must be shared:

Shared tables	
table code	table name
tcom000	Companies

Data can be shared across group companies and between stand alone companies (single logistic/ single finance), as well as within a group company.

If data sharing is required across group companies or stand alone companies for any table with a **Finance Company** field (i.e. warehouse, work center, projects, customers, suppliers, first free numbers, purchase, sales and production orders), the tables can not be logically linked. As well, caution is advised while building replication schemes to properly determine the value for the Finance Company field.

13.2 Multi Logistic/ Single Finance

In this structure financial integration transactions from all logistic companies will be posted to only one finance company, therefore no intercompany transactions occur. The group company is the finance company used. All tables in the finance package must be shared, as well as other tables used by both finance and logistic companies. The non financial tables do not have to be shared entirely, however the group company must contain the data that exists in each logistic company.

Shared modules	
Module code	Module name
ACP	Accounts Payable
ACR	Accounts Receivable
CAL	Cost Allocation
CMG	Cash Management
FAS	Fixed Assets
FBS	Financial Budgeting
FST	Financial Statements
GLD	General Ledger
MCS	Tables all except the following tables: tcmcs095 – Parameters tcmcs096 – Statuses tcmcs097 - Performance Boosters tcmcs097 - Performance Boosters
COM	Common**

** The data in these tables do not have to be entirely shared between all companies, however, the group company (the finance company) must contain all data used in the Finance package for all companies. Any data not used in the Finance package does not need to be shared with the group company (i.e. forwarding agents).

The text tables (TXT module) in the Tools package (TT) must be shared.

Baan IVc allows central accounts payable invoice matching without sharing the purchase order tables. If this functionality is not used the following purchase order tables must be shared:

Shared tables	
table code	table name
tdpur040	Purchase Orders
tdpur041	Purchase Order Lines
tdpur045	Receipts
tdpur046	Purchase Invoices

If the above tables are shared, it is recommended, but not mandatory, to share the following tables:

Shared tables	
table code	table name
tdpur042	Specific Addresses (Purchase Orders)
tdpur047	Purchase Order Bills of Material
tdpur048	Purchase Receipts for Posting Work in Process
tdpur050	Purchase Order History
tdpur051	Purchase Order Line History

13.3

Single Logistic/ Multi Finance

In this structure financial integration transactions from the logistic company will be posted to the finance companies specified on the transaction. Intercompany transactions will occur when the finance companies on the debit and credit entries of the transaction are different. The finance company can be specified on the warehouse, work center, project, as well as on the sales, purchase and production orders.

The logistic company used for processing must be the same company number as the group company. All logistic company tables must be shared, as well as some finance tables.

All tables in the following packages must be shared:

Shared packages	
package code	package name
TC	Common **
TD	Distribution
TI	Manufacturing
TP	Project
TR	Transportation
TS	Service

** The data in these tables do not have to be entirely shared between all finance companies, however, the group company must contain all data used in the Finance package for all companies.

The text tables (TXT module) in the Tools package (TT) must be shared.

The following finance tables must also be shared:

Shared tables	
table code	table name
tfacp001	Financial Supplier Groups
tfacp004	Aging Analysis Data
tfacp005	Ledger Accounts by Supplier Group
tfacr001	Financial Customer Groups
tfacr005	Ledger Accounts by Customer Group
tfacr310	Reminder Methods
tfacr311	Statement of Account Methods
tfacr312	Statement of Account Layouts
tfcmg003	Payment / Receipt Methods
tfgld000	Logistic Companies by Finance Company
tfgld001	Companies by Group Company
tfgld002	Dimension Descriptions
tfgld003	Group Company Parameters
tfgld005	Periods
tfgld006	End Dates by Year
tfgld015	InterCompany Relations
tfgld102	Non Finalized Transactions
tfgld107	Interim Tax
tfgld110	Tax Analysis
tfgld111	Interim Tax Adjustments
tfgld112	Relation InterCompany cash Documents
tfgld201	History - ledger Account Totals
tfgld202	History - Dimension/Ledger Account Totals
tfgld203	Opening Balances - Ledger/Currency
tfgld204	Opening Balances - Dimension/Ledger/Currency
tfgld205	History - Dimension/Ledger Account Combination Totals
tfgld206	Opening Balances - combination of Dimension/Ledger/Currency
tfgld410	Integration Transactions
tfgld411	Tax Distribution for Integration Transactions
tfgld412	Invoice Details for Integration transactions
tfgld413	Receipts Schedule for Integration Transactions
tfgld414	One Time Addresses for Integration Transactions
tfgld415	Additions to Integration Transactions
tfgld416	Non Posted Integration Transactions
tfgld417	Posted Integration Transactions
tfgld418	Integration References
tfgld419	Logistic Data
tfgld420	Integration Elements

tfgld421	Options by Integration Element
tfgld422	Simple Integration Relations by Range
tfgld423	Simple Integration Relations by Elements
tfgld424	Composed Integration Relations by Range
tfgld425	Composed Integration Relations by Element
tfgld426	Transaction Types by Transaction Origin
tfgld432	Integration Transactions Log
tfgld433	Non Finalized Transaction Origins
tfgld450	Integration Parameters

13.4

Multi Logistic/ Multi Finance

In this structure the tables that require sharing will vary depending on the situation. Because the group company functionality is finance functionality, the following finance tables must be shared, regardless of the chosen structure. The use of the functionality, as well as data management requirements will determine how/if other tables will be shared.

Shared tables	
table code	table name
tccom000	Companies
tfgld000	Logistic Companies by Finance Company
tfgld001	Companies by Group Company
tfgld002	Dimension Descriptions
tfgld003	Group Company Parameters
tfgld005	Periods
tfgld006	End Dates by Year
tfgld015	InterCompany Relations
tfgld102	Non Finalized Transactions
tfgld107	Interim Tax
tfgld110	Tax Analysis
tfgld111	Interim Tax Adjustments
tfgld112	Relation InterCompany cash Documents
tfgld201	History - ledger Account Totals
tfgld202	History – Dimension/Ledger Account Totals
tfgld203	Opening Balances - Ledger/Currency
tfgld204	Opening Balances - Dimension/Ledger/Currency
tfgld205	History – Dimension/Ledger Account Combination Totals
tfgld206	Opening Balances - combination of Dimension/Ledger/Currency

The next chapters discuss table sharing options in the Manufacturing, Distribution and Common packages.

14.

BAAN IVc

The information in this appendix is valid for BAAN version IVc4. Refer to 'Data sharing by package' in Chapter 8 for an explanation of the information in this appendix.

14.1

Common Data (TC)

Employees	
table code	table name
tccom001	Employees

Customers	
table code	table name
tccom010	Customers
tccom011	Customer Balances by Company
tccom012	Customer postal Addresses
tccom013	Delivery Addresses
tccom015	Banks by Customer

Suppliers	
table code	table name
tccom020	Suppliers
tccom021	Supplier Balances by Company
tccom022	Supplier Postal Addresses
tccom025	Banks by Supplier

Note:

The balance and turnover fields in the customer and supplier files. This dynamic data may or may not be shared depending on the business requirements.

Logistic and Financial Tables	
table code	table name
tcmcs001	Units
tcmcs002	Currencies
tcmcs003	Warehouses
tcmcs004	Routes
tcmcs005	Reasons for Rejection
tcmcs006	Unit Sets
tcmcs007	Units by Language
tcmcs008	Currency Rates
tcmcs009	Credit Insurance Companies
tcmcs010	Countries
tcmcs011	Late Payment Surcharges
tcmcs012	Units by Unit Set
tcmcs013	Terms of Payment
tcmcs014	Payment Schedules
tcmcs015	Product Types
tcmcs016	Seasonal Patterns
tcmcs018	Signals
tcmcs019	Titles
tcmcs020	Bank Addresses
tcmcs021	Discount Codes
tcmcs022	Selection Codes
tcmcs023	Item Groups
tcmcs024	Price Groups
tcmcs025	Factoring Companies
tcmcs026	Banks per Factoring Company
tcmcs027	Item Code Systems
tcmcs028	Commodity Codes
tcmcs029	Quality Groups
tcmcs030	Skills
tcmcs031	Lines of Business
tcmcs032	Single Tax Rates
tcmcs033	Multiple Tax Rates
tcmcs034	Price Lists
tcmcs035	Tax Rates by Tax Code
tcmcs036	Tax Codes by Country
tcmcs037	Tax Codes
tcmcs039	Order Steps
tcmcs041	Terms of Delivery
tcmcs042	Order Types
tcmcs043	Additional Cost Sets

tcmcs044	Statistics Groups
tcmcs045	Areas
tcmcs046	Languages
tcmcs047	First Free Numbers **
tcmcs053	Rounding Codes
tcmcs055	Invoicing Methods
tcmcs080	Forwarding Agents

** If first free numbers are not shared, it can be necessary to ensure unique series per company under certain circumstances.

14.2

Distribution (TD)

Purchase Orders	
table code	table name
tdpur040	Purchase Orders
tdpur041	Purchase Order Lines
tdpur045	Receipts
tdpur046	Purchase Invoices

If the above tables are shared, it is recommended, but not mandatory, to share the following tables:

Purchase Orders	
table code	table name
tdpur042	Specific Addresses (Purchase Orders)
tdpur047	Purchase Order Bills of Material
tdpur048	Purchase Receipts for Posting Work in Process
tdpur050	Purchase Order History
tdpur051	Purchase Order Line History

Note:

If the above purchase order tables are shared and subcontract purchase orders are used, the following tables must also be shared.

Purchase Orders	
table code	table name
ticst002	Estimated and Actual Hours
tisfc001	Production Orders

Purchase Statistics Set Up Data

table code	table name
tdpst096	Purchase Statistics Layout Codes
tdpst097	Column Content by Layout Code
tdpst098	Purchase Statistics Display Codes
tdpst200	Specific Selection on Country
tdpst201	Specific Selection on Price List
tdpst202	Specific Selection on Buyer
tdpst204	Specific Selection on Supplier
tdpst205	Specific Selection on Address Code
tdpst206	Specific Selection on Warehouse
tdpst207	Specific Selection on Statistics Group
tdpst208	Specific Selection on Item Group
tdpst209	Specific Selection on Project
tdpst210	Specific Selection on Item
tdpst211	Specific Selection on Area
tdpst212	Specific Selection on Line of Business
tdpst213	Specific Selection on Order Type
tdpst214	Specific Selection on Supplier Group
tdpst215	Specific Selection on Discount Code
tdpst219	Specific Selection on Financial Company

Purchase Statistics Transaction Data	
table code	table name
tdpst010	Purchase Statistics Master File (Turnover)
tdpst015	Purchase Statistics Master File (Order Intake)
tdpst018	Purchase Statistics Master File (Cancellations)
tdpst020	Purchase Statistics Compression (Turnover)
tdpst025	Purchase Statistics Compression (Order Intake)
tdpst028	Purchase Statistics Compression (Cancellations)
tdpst030	Purchase Statistics Sort File (Key File)
tdpst050	Purchase Budgets
tdpst090	Purchase Statistics Period Data
tdpst094	Purchase Statistics Sort Codes
tdpst095	Purchase Statistics Compression
tdpst110	Purchase Statistics Discount Type (Turnover)
tdpst115	Purchase Statistics Discount Type (Order Intake)
tdpst118	Purchase Statistics Discount Type (Cancellation)
tdpst120	Purchase Statistics Compression for Discount Type (Turnover)
tdpst125	Purchase Statistics Compression for Discount Type (Order Intake)
tdpst128	Purchase Statistics Compression for Discount Type (Cancellation)
tdpst150	Purchase Budgets by Discount Type

Purchase Prices and Discounts	
table code	table name
tdpur030	Prices by Supplier and Item
tdpur031	Discounts by Supplier and Price Group
tdpur032	Prices by Price List and Item
tdpur033	Discounts by Prices List and Price Group
tdpur034	Discount Parameters by Supplier and Price Group
tdpur035	Discount Parameters by Price List and Price Group
tdpur036	Price Parameters by Supplier and Item
tdpur037	Price Parameters by Price List by Item

Purchase Contracts	
table code	table name
tdpur300	Purchase Contracts
tdpur301	Purchase Contract Lines
tdpur303	Specific Addresses (Purchase Contracts)
tdpur305	Purchase Contract History
tdpur306	Purchase Contract Line History
tdpur332	Purchase Contract Prices
tdpur333	Purchase Contract Discounts
tdpur335	Purchase Contract Discount Parameters
tdpur336	Purchase Contract Price Parameters

Sales Orders	
table code	table name
tdsls038	Sales Order Margin Control Log File ** optional
tdsls040	Sales Orders
tdsls041	Sales Order Lines
tdsls045	Sales Deliveries
tdsls049	Delivered Sales Orders to be Processed

If the above tables are shared, it is recommended, but not mandatory, to share the following tables:

Sales Orders	
table code	table name
tdsls042	Specific Addresses (Sales Orders)
tdsls043	Bill of Lading Data (Sales Orders)
tdsls044	Unblocked Sales Orders
tdsls047	Sales Order Bills of Material
tdsls048	Installment Schedules
tdsls050	Sales Order History
tdsls051	Sales Order Line History

Sales Statistics Set Up Data	
table code	table name
tdsst096	Sales Statistics Layout Codes
tdsst097	Column Content by Layout Code
tdsst098	Sales Statistics Display Codes
tdsst200	Specific Selection on Country
tdsst201	Specific Selection on Price List
tdsst202	Specific Selection on Sales Representative
tdsst203	Specific Selection on Parent Customer
tdsst204	Specific Selection on Customer
tdsst205	Specific Selection on Address Code
tdsst206	Specific Selection on Warehouse
tdsst207	Specific Selection on Statistics Group
tdsst208	Specific Selection on Item Group
tdsst209	Specific Selection on Project
tdsst210	Specific Selection on Item
tdsst211	Specific Selection on Area
tdsst212	Specific Selection on Line of Business
tdsst213	Specific Selection on Order Type
tdsst214	Specific Selection on Customer Group
tdsst215	Specific Selection on Discount Code
tdsst216	Specific Selection on Commission
tdsst217	Specific Selection on Rebate
tdsst218	Specific Selection on Contact
tdsst219	Specific Selection on Financial Company

Sales Statistics transaction Data

table code	table name
tdsst010	Sales Statistics Master File (Turnover)
tdsst015	Sales Statistics Master File (Order Intake)
tdsst018	Sales Statistics Master File (Cancellations)
tdsst020	Sales Statistics Compression (Turnover)
tdsst025	Sales Statistics Compression (Order Intake)
tdsst028	Sales Statistics Compression (Cancellations)
tdsst030	Sales Statistics Sort File (Key File)
tdsst050	Sales Budgets
tdsst090	Sales Statistics Period Data
tdsst094	Sales Statistics Sort Codes
tdsst095	Sales Statistics Compression
tdsst110	Sales Statistics Discount Type (Turnover)
tdsst115	Sales Statistics Discount Type (Order Intake)
tdsst118	Sales Statistics Discount Type (Cancellation)
tdsst120	Sales Statistics Compression for Discount Type (Turnover)
tdsst125	Sales Statistics Compression for Discount Type (Order Intake)
tdsst128	Sales Statistics Compression for Discount Type (Cancellation)
tdsst150	Sales Budgets by Discount Type

Sales Prices and Discounts

table code	table name
tdsls030	Prices by Customer and Item
tdsls031	Discounts by Customer and Price Group
tdsls032	Prices by Price List and Item
tdsls033	Discounts by Price List and Price Group
tdsls034	Discount Parameters by Customer and Price Group
tdsls035	Discount Parameters by Price List and Price Group
tdsls036	Price Parameters by Customer per Item
tdsls037	Price Parameters by Price List by Item

Sales Contracts	
table code	table name
tdsls300	Sales Contracts
tdsls301	Sales Contract Lines
tdsls303	Specific Addresses (Sales Contracts)
tdsls305	Sales Contract History
tdsls306	Sales Contract Line History
tdsls332	Sales Contract Prices
tdsls333	Sales Contract Discounts
tdsls335	Sales Contract Discount Parameters
tdsls336	Sales Contract Price Parameters

14.3

Manufacturing (TI)

Items	
table code	table name
tiitm001	Items
tiitm002	Item Default Data
tiitm003	Alternative Units
tiitm004	Conversion Factors
tiitm011	Alternative Items

Bills of Material	
table code	table name
tibom010	Production BOMs

Routings	
table code	table name
tirou001	Work Centers
tirou002	Machines
tirou003	Tasks
tirou030	Norm Time Tables
tirou031	Norm Times
tirou101	Routing Codes by Item
tirou102	Routing
tirou420	Day Types

Cost Accounting Set Up Data

table code	table name
ticpr010	Cost Price Components
ticpr050	Operation Rate Codes
ticpr100	Cost Price Calculation Codes
ticpr110	Cost Price Surcharges by Item Group
ticpr120	Cost Price Surcharge Bases by Item Group
ticpr130	Cost Price Surcharges by Item
ticpr140	Cost Price Surcharge Bases by Item
ticpr150	Operation Rates
ticpr160	Subcontracting Rates
ticpr170	Simulated Purchase Prices

Cost Accounting Data by Item

table code	table name
ticpr200	Cost Prices by Item
ticpr210	Items by Cost Price Calculation Code
ticpr220	Cost Price Surcharges by Item and Cost Price Component
ticpr230	Cost Price Surcharge Bases by Item and Cost Price Component
ticpr300	Standard Cost Prices by Item
ticpr320	Cost Price Surcharges by Item and Effective Date
ticpr330	Cost Price Surcharge Bases by Effective Date

Engineering Data Management Set Up Data

table code	table name
tiedm001	Engineering Item Groups
tiedm002	Drawing Sizes
tiedm003	Drawing Locations

Engineering Data Management item Data

table code	table name
tiedm010	Engineering Items
tiedm100	Engineering Revisions
tiedm101	Engineering Item / Item Relationships
tiedm110	Engineering BOMs
tiedm120	EBOM Copy Data

Engineering Change Order Set Up Data

table code	table name
tiedm301	ECO Procedures
tiedm302	ECO Procedure Steps
tiedm303	Steps by ECO Procedure
tiedm304	ECO Reasons

Engineering Change Order Transaction Data

table code	table name
tiedm310	ECOs
tiedm311	ECO Procedure Steps By ECO
tiedm320	ECO Lines
tiedm321	Included/Excluded E-items by ECO
tiedm322	E-items by ECO

Hours Accounting Master Data

table code	table name
tihra110	Working Time Tables
tihra120	Working Hours
tihra130	Hourly Labour Types
tihra300	Period Table

Product Configuration Set Up Data	
table code	table name
tipcf050	Product Features
tipcf051	Product Feature Descriptions
tipcf060	Product Feature Options
tipcf061	Option Descriptions by Product Feature
tipcf100	Product Features by generic Item
tipcf101	Product Feature Descriptions by Generic Item
tipcf110	Options by Product Feature and Generic Item
tipcf111	Option Descriptions by Product Feature and Generic Item
tipcf200	Constraint Ids by Generic Item
tipcs210	Constraints by Generic Item
tipcf220	Constraint Validation Messages
tipcf300	Generic Item Data
tipcf310	Generic BOMs
tipcf320	Generic Routing
tipcf400	Generic Price Lists
tipcf401	Generic Price List Descriptions
tipcf410	Price List Matrix Ids
tipcf420	Price List Matrix

PCS Data	
table code	table name
tipcs020	Projects

14.4

Finance (TF)

General Ledger Master Data	
table code	table name
tfgld008	Chart of Accounts
tfgld009	Dimension Ranges by Ledger Account
tfgld010	Dimensions
tfgld011	Transaction Types
tfgld012	Transaction Schedules
tfgld013	Transaction Schedule Details
tfgld019	Currency Difference Ledger Accounts by Currency
tfgld035	Protocol Codes

Financial Integration Set Up	
table code	table name
tfgld420	Integration Elements
tfgld421	Options by Integration Element
tfgld422	Simple Integration Relations by Range
tfgld423	Simple Integration Relations by Elements
tfgld424	Composed Integration Relations by Range
tfgld425	Composed Integration Relations by Element
tfgld426	Transaction Types by Transaction Origin
tfgld450	Integration Parameters
tdinv850	Inventory and WIP Transaction Accounts
tdsls070	Sales Invoice Analysis Accounts
tdsls071	Sales Invoice Details Accounts
tipcs850	Chart of Accounts for Financial Project Transactions

Financial Integration Transactions	
table code	table name
tfgld410	Integration Transactions
tfgld411	Tax Distribution for Integration Transactions
tfgld412	Invoice Details for Integration transactions
tfgld413	Receipts Schedule for Integration Transactions
tfgld414	One Time Addresses for Integration Transactions
tfgld415	Additions to Integration Transactions
tfgld416	Non Posted Integration Transactions
tfgld417	Posted Integration Transactions
tfgld418	Integration References
tfgld419	Logistic Data
tfgld432	Integration Transactions Log
tfgld433	Non Finalized Transaction Origins

Accounts Payable Master Data	
table code	table name
tfacp001	Financial Supplier Groups
tffacp002	Blocking Codes
tfacp003	Blocking References
tfacp004	Aging Analysis Data
tfacp005	Ledger Accounts by Supplier Group
tfacp006	Tolerated Price Differences by User
tfacp306	Subcontracting Payments by Payment Batch

If the ACP parameter for central invoicing is set to No, then the following purchase tables must be shared:

Central Accounts Payable Invoicing	
table code	table name
tdpur040	Purchase Orders
tdpur041	Purchase Order Lines
tdpur045	Receipts
tdpur046	Purchase Invoices

If the above tables are shared, it is recommended, but not mandatory, to share the following tables:

Central Accounts Payable Invoicing

table code	table name
tdpur042	Specific Addresses (Purchase Orders)
tdpur047	Purchase Order Bills of Material
tdpur048	Purchase Receipts for Posting Work in Process
tdpur050	Purchase Order History
tdpur051	Purchase Order Line History

If the above purchase order tables are shared and subcontract purchase orders are used, the following tables must also be shared.

Central Accounts Payable Invoicing

table code	table name
ticst002	Estimated and Actual Hours
tisfc001	Production Orders

Accounts Receivable Master Data

table code	table name
tfacr001	Financial Customer Groups
tfacr002	Problem Codes
tfacr003	Problem References
tfacr005	Ledger Accounts by Customer Group
tfacr300	Reminder Variables
tfacr301	Reminder Letters
tfacr302	Reminder Costs
tfacr310	Reminder Methods
tfacr311	Statement of Account Methods
tfacr312	Statement of Account Layouts
tfacr313	Statement of Account Letters
tfacr314	Statement of Account Variables
tfacr315	Statement of Account Texts

Cost Allocation Master Data

table code	table name
tfc001	Cost Allocations
tfc002	Cost Drivers
tfc003	Cost Categories
tfc004	Cost Objects
tfc005	Activities
tfc006	Ledger Accounts by Cost Category
tfc100	Allocation Sources
tfc101	Allocation Targets
tfc102	Fixed Quantities (Allocation Sources)
tfc103	Fixed Quantities (Target Allocations)
tfc200	Bill of Activities
tfc201	Surcharge Base Quantities

Cash Management Master Data

table code	table name
tfcmg001	Bank Relations
tfcmg002	Reasons for Foreign Payments
tfcmg003	payment/Receipt Methods
tfcmg006	Data by Bank & Payment/Receipt Method
tfcmg007	Posting Data by bank & Payment/Receipt Method
tfcmg008	Posting Data by tax Code
tfc306	Subcontracting Payments by Payment Batch

Fixed Assets Master Data	
table code	table name
tffas002	Depreciation Master Data
tffas003	Fixed Asset Main Groups
tffas004	Fixed Asset Groups
tffas005	group Depreciation Data
tffas007	Fixed Asset Master Data
tffas008	Fixed Asset Depreciation Data
tffas009	Fixed Asset Variable Depreciation Data
tffas013	Fixed Asset Locations
tffas014	Business Information Codes
tffas015	Business Information Types
tffas016	Business Information Definitions
tffas017	Business Information Details
tffas018	Special Depreciation Master Data
tffas019	Special Depreciation Amounts

Fixed Asset Transaction Data **	
table code	table name
tffas100	Investments
tffas101	Disposals
tffas102	Maintenance Costs

** Do not share tffas020 - Logged Transactions

Financial Statements (FST)	
table code	table name
tffst010	Statement Layouts
tffst011	Statement Column Layouts
tffst012	Statement Account Layouts
tffst013	Statement Account Column Layouts
tffst020	Statement Annexure Layouts
tffst021	Annexure Column Layouts
tffst022	Annexure Account Layouts
tffst023	Annexure Account Column Layouts
tffst030	Consolidated Statement Layouts
tffst031	Consolidated Statement Column Layouts
tffst032	Consolidated Statement Account Layouts
tffst033	Consolidated Statement Account Column Layouts
tffst100	Financial Statements
tffst110	Statement Columns
tffst111	Statement Column Variable References
tffst120	Statement Accounts
tffst121	Statement Ledger/Dimension Accounts
tffst122	Statement Ledger/Dimension Accounts by Column
tffst123	Statement Account/Column Specifications
tffst124	Statement Account Ration Specifications by Column
tffst125	Statement Alternate Account Specifications
tffst130	Alternate Financial Statement Account Groups
tffst400	Consolidated Statements
tffst410	Consolidated Statement Columns
tffst411	Consolidated Statement Column References

Whenever data needs to be updated in the database, the B-shell must also update the audit log files. These are two actions, that are performed in sequence. However when during one of these actions the network, the db-driver or the audit server goes down, this results in discrepancies between the records in the database and the logged entries in the audit log records.

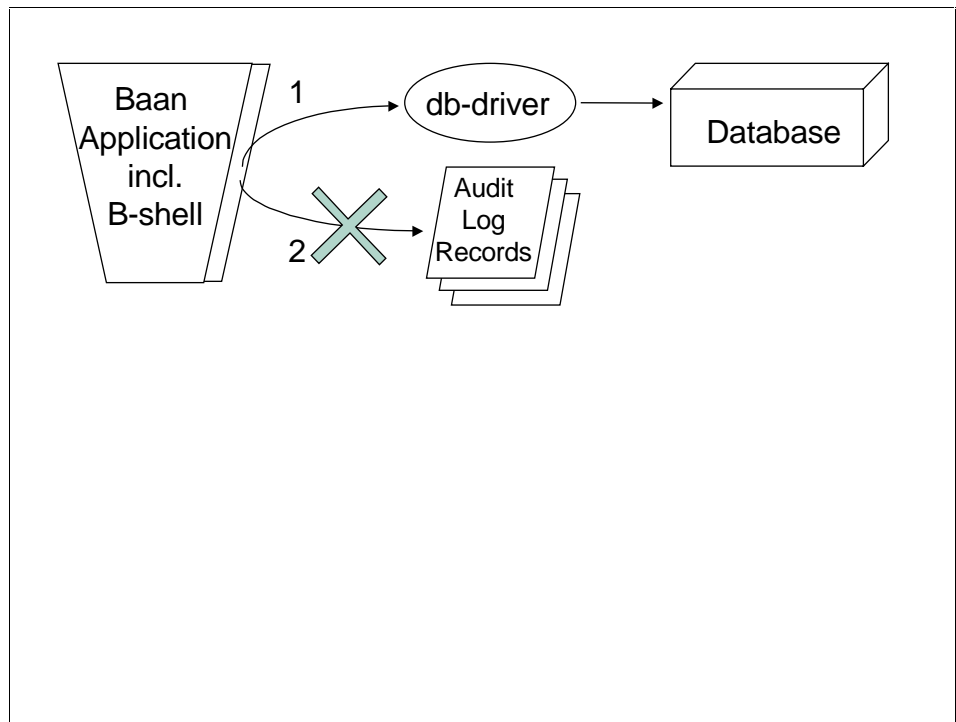


Figure 31 In-doubt transactions

If there is a parent child relation between records the reference counters are used. Reference counters are used in the parent records and indicate how many children the parent has. Every time a child record is added or deleted the reference counter is increased or decreased.

In the application logic different reference types can be defined by parent/child relation. The reference type consists of the reference mode and the reference action. The reference mode describes whether or not the parent child relation is compulsory.

The reference action describes which action is taken upon a delete or update of the parent. The reference action 'look up' means that if of a delete of the parent, first all child records are checked for possible links to this parent. The reference action 'restricted', means that on deletion of the parent record only the reference counter is checked whether or not there are any child records. It is clear that the 'restricted' action gives a much better performance in deletion actions.

Using Exchange the reference counters are updated also by the B-shell. However replicating reference counters makes only sense when all child tables are replicated along. When a database generated replication method is used the reference counter are not updated, , which may result in inconsistent counters when child records are added or deleted (See the example). This can be avoided by having a master-slave relations between instances of data and a good control of data ownership.

In the Baan application most of the reference actions are by default 'look up'.

Example

A sales order line must have a reference to a sales order header. The relation is compulsory unless the sales order line is empty (reference mode). For performance reasons the reference action 'restricted' is chosen for this parent/child relation.

If on the local server a sales order header with 3 sales order lines is generated. The reference counter on local database will read 3. Upon replication by means of database replication the reference counter is not replicated. In the central server the reference counter on the sales order header will read zero. When on the central server the instruction 'delete sales order header' is given the system will check the reference counter. This will read zero, and therefore the header can be deleted. In real there are sales order lines present. This results in inconsistent data on the central server.

Every indexed set of database fields has a hash column. The column can not be seen by the user and is generated by the database driver for performance reasons e.g. during queries. A hash column value consists of the combined values of its indexed fields.

When replication is done via Exchange or via the database driver, the hash columns on the remote server are updated also. When data is replicated by the database system this is not the case. This may lead to inconsistent database record values in the replicated records.

This can be avoided by replicating all indexed columns and the hash column, and by observing a master-slave relation between the records.

Example

On the local server the indexed fields read the following values: "01", "634" and "F5". The hash column will have the value "01634F5".

When the second column, the third column and the hash column are replicated, by means of database replication, the table columns on the central server may read there: "02", "634" and "F5". The hash column will read "**01**634F5".

When Exchange is for replication, the data in the first column can be transformed upon import into the central server. In this case the value "01" would be replaced by "02". The Exchange module would also take care of the correction in the hash column.

Listed below is the possible input data needed from local MPS plans. Which data exactly is needed, depends on several parameter values. E.g. Rebuild Supply Chain, Aggregate Demand Forecast, Aggregate Demand Plan, Aggregate Production Plan, etc..

- Plan Sites
- Interplant Supplying Sites
- Interplant Supplying Priorities
- Plan Periods
- Company Calendar
- Workcenter Calendars
- Planner Authorisation
- Bill of Critical Materials
- Bill of Critical Capacities
- Plan Code e.g. Demand History Parameters values, Backward/Forward Consumption parameter values
- Plan Items e.g. Forecasting parameters values, Safety Stock level
- Item Data e.g. order parameter values, lead time
- MPS Parameters e.g. Max. number of orders per day, Smoothing Factors for Rescheduling Messages, Demand & Inventory Exception Messages-boundaries
- MPS Plan Data (of the previous plan):
 - Demand Forecast
 - Extra demand
 - Customer Orders
 - Customer Deliveries

- Goods Flow Information
 - Demand Forecast (B)
 - Extra Demand (B)
 - Customer Orders (A, B, C)
 - Exploded Demand (A, B, C)
 - Exploded Rough Material demand (A, B, C)
 - Exploded Interplant demand (A, B, C)
 - Customer Deliveries (A, B, C)
 - Internal Deliveries (A, B, C)
 - Internal Interplant Deliveries (A, C)
 - Planned Receipts (B)
 - Scheduled Receipts (A, B, C)
 - Actual Receipts (A, B, C)
 - Planned Interplant Receipts (A, B, C)
 - Scheduled Interplant Receipts (A, C)
 - Actual Interplant Receipts (A, C)
 - Planned Inventory (B)
 - Actual Inventory (A, C)

where:

A = plan item is product family of higher plan level (multisite plan & local plan)

B = plan item is product family or MPS item of lowest plan level (multisite plan)

C = plan item is MPS item (local plan)

The following sessions must be executed to set up a company. The group company must always be set up first. Except as noted, follow this procedure to set up all other companies in the group. In the case of more than one group company, the group company that is the base company must be set up first.

- 1 Maintain Companies (ttaad1100m000)
- 2 Assign Tables to Database (ttaad4111m000)
- 3 Maintain Table Directories (ISAM) (ttaad4115m000)
- 4 Maintain Logical Tables (ttaad4120m000) ****A**
- 5 Conversion to Runtime Data Dictionary (ttaad4200m000)
- 6 Create Tables (ttaad4230m000)
- 7 Check Tables (ttaad4232m000)
- 8 Maintain Company Data (tcom0100m000) ****B**
- 9 Maintain Countries (tcmcs0110m000)
- 10 Maintain Languages (tcmcs0146m000)
- 11 Maintain Currencies (tcmcs0102m000)
- 12 Maintain Currency Rates (tcmcs0108m000)
- 13 Maintain Company Parameters (tfgld0103m000)
- 14 Maintain Group Company Parameters (tfgld0101m000) ****C**
- 15 Maintain Periods (tfgld0105m000) ****D**
- 16 Maintain Period Status (tfgld0107m000)

A: Logical table linking is not used on the first company that is set up, but will be used on subsequent companies.

B: The four sessions following must be maintained within the session Maintain Company Data, and may be accessed by zooming on the fields.

C: Group company parameters are only maintained for group companies (all companies within the group use the group company's data). Maintaining this session is not required for any other companies within the group.

D: The field Current Year may now be filled in the session Maintain Company Parameters.

Other company data may be set up as required once this base data is set up. Note that logical table linking and/or other synchronization methods must be in place prior to setting up related data.

