BAAN IV b

New Functions in BAAN IV b

Functional Release Information P3000C US

Research and Development

Document information

Document

Number : P3000C US
Type : Development Document
Department : Research & Development
Name : New Functions in BAAN IV b
Version : C
Date : December

Table of Contents

About this Document

Highlights of BAAN IV b

This document describes the new functionality of BAAN IV b as opposed to BAAN IV a.

The main change in BAAN IV b is the support for hybrid production situations with MPS/MRP for both non-Process and Process items, based on multilevel BOMs and formula structures. This functionality is designed for situations where Process items and non-Process items are produced in combination.

BAAN IV b contains Intelligent Requirements Planning (IRP), which provides most of the MPS functionality required for Process. The release of IRP combines constraint cyclic and traditional infinite MPS planning (with coproducts).

The BAAN IV b development focus for Electronic Data Interchange (EDI) and Distributed Data Collection (DDC) was primarily to develop the following standard requirements:

- Reconstruct the EDI technology base
- Enhancements to selected EDI transactions
- Moving DDC functionality from the GLO localization to the standard BAAN package.

Several performance improvements have been made to BAAN IV b. In BAAN Tools, the Windows interface, bshell and the standard program have been optimized. For applications, it is possible to spread the workload of critical batch programs across the available processor capacity. This is possible, for example, for a Material Requirements Plan (MRP) run.

1 Common

1.1 Electronic Data Interchange (EDI)

The BAAN IV b release includes a number of enhancements to the standard EDI module. These enhancements provide improved functionality in the processing of incoming EDI messages, and fall into the following three categories:

- Error handling
- Data preview and manipulation
- Single file layout option

The underlying process by which documents are imported into BAAN IV b, has been modified to improve the processing, viewing, and manipulation of incoming EDI documents. The module now supports the following functional requirements.

Error handling

Improved error handling provides users with the ability to easily view, correct, and resubmit incoming documents processed with errors, using BAAN IV b sessions. Facilities for online error review, and online data review and correction are available.

Previewing incoming data

Users have the option to view, by trading partner and message type, incoming documents prior to the EDI processing. Optionally, users can also view and manipulate incoming data, via BAAN IV b sessions, prior to bringing documents into the BAAN IV b base application. If this option is not used, documents will be processed and updated into the BAAN IV b base application directly, without user intervention.

Single file layout option

In earlier BAAN versions, the EDI module required incoming documents to be split across multiple files (one for header data, one for line item data, and so on). In many cases, however, third-party EDI translators used in conjunction with the EDI module expect all data associated with a particular document to be contained in a single file. In order to provide maximum flexibility, BAAN IV b supports both single and multiple file layouts.

The matrix below shows all the messages supported in BAAN IVb:

Message by package	Module	Direction	
		In	Out
Distribution			
Purchase Order	PUR		x
Sales Order	SLS	x	
Sales Acknowledgement	SLS		x
Purchase Acknowledgement	PUR	x	
Despatch Advice	SLS		x
Despatch Advice	PUR	x	
Sales Invoice	SLS		x
Finance			
Purchase Invoice	ACP	x	
Remittance Advice	CMG	x	
Transportation			
Transportation Order	TOC	x	
Transportation Order Acknowl.	TOC		x
Inbound Order	TOC	x	
Inbound Order Acknowledgement	TOC		x
Outbound Order	TOC	x	
Outbound Order Acknowledgement	TOC		x
Supply Chain			
Purchase Schedule	PSC		x
Sales Schedule	SSC	x	
Sales Shipping Schedule	SSC		x
Purchase Shipping Schedule	PSC	x	
Advanced Shipping Note	SSC		x
Advanced Shipping Note	PSC	x	
Application Advice	PSC		x
Application Advice	SSC	x	
Receiving Acceptance Advice	PSC		x
Receiving Acceptance Advice	SSC	x	

The current structure of the EDI-module provides the user the flexibility to add new messages in an easy way.

Each combination of a message and a direction requires a specific program (dll). For a new message only a new dll needs to be developed, which can be included in the EDI structure without changes in the EDI-module itself.

1.2 Performance boosters

Several boosters have been introduced to influence the performance of particular sessions. Totally new are table boosters to set tables the user wants to load into cache memory. This kind of reading data will provide fast data access and result in a considerable performance improvement.

In addition, performance boosters can be maintained. A new feature is that sessions can be run using multiple bshells, which is enabled by the new performance booster Number of Servers.

2 Distribution

2.1 Commission and Rebate Management System (CMS)

Commissions paid to employees

In BAAN IV b, a business object is added to the CMS module in order to handle commission payments to employees (sales representatives). Calculating Commissions To Employees (CTE) was already possible in BAAN IV a, but the procedure to handle the actual payment was not yet available. Since actual payment of CTE is not done by invoicing procedures but via payroll payments, this additional business object manages the financial aspects of CTE.

After the CTEs have been calculated and financial reservations made, a report can be generated, which shows the (re)calculated CTE. This report can be handed over to the financial controller handling the payroll transactions. After actual payment of commissions by means of payroll transactions (debiting the correct ledger account for reserved CTEs), the financial controller informs the CMS controller about these payments by passing on the listing of (re)calculated CTEs. This is where the new business object starts. The CMS controller can register the actual payments with it. Within this business object a balance report shows the balance of calculated CTEs and, after handling this payment procedure, the balance amount will be equal to the ledger account balance in Finance indicating the reserved CTEs for a particular fiscal year and period. Paid and (partially) unpaid commission are made visible in this business object.

2.2 Purchase and Sales Statistics (PST/SST)

BAAN IV b includes the following enhancements to the purchase and sales statistics:

Random selections for all statistics levels

In previous releases it was only possible to print consecutive ranges of information. In BAAN IV b it is also possible to specify random, nonconsecutive selections. In the layout codes it is possible to define a nonconsecutive selection of customers for printing statistical information.

Sublevel of discount code

In BAAN IV a, multiple discounts per order line was introduced. It was possible to link a discount code to each discount level. BAAN IV b offers the possibility to define budgets and to compose reports based on discount codes. In budgeting, a total discount amount has to be defined, which can be divided over various discount codes.

The print sessions allow the user to select on discount level and to print a specification per discount code.

Full update using actual master data

(Full) update of statistics can now be done either on the basis of historical data or based on the actual data in the master tables. For each statistics level (country, customer, and so on.) it is possible to choose whether the update should be based on historical data or on actual data.

2.3 Interfaces

Interface between Sales Statistics (SST) and Commissions (CMS)

In BAAN IV a, the commission and rebate (CMS) module was added to the standard package. In the BAAN IV b release, the interface between sales statistics and the commissions/rebates module is new. This makes it possible to present commission and rebate information together with statistical information. Two new entries (commission amount and rebate amount) are available, in addition to the existing entries (amount, quantity, alternative quantity, discount, cost of sales, and gross profit). All functions for the existing levels are also available for the new entries.

2.4 Changes to Standard Code for Multisite

Introduction

Baan has made several changes to standard multisite functionality. Below you find a description of the modifications, as well as background information on why the changes were made. In most cases the changes were required to accommodate multiserver situations.

The changes are:

- Central purchase contracts with decentral purchase orders.

- Multiserver customer and supplier changes.

Replication

There are numerous references to replication of data in the following descriptions. Baan does not provide synchronous replication tools for these specific changes. Decisions must be made within each implementation as to the appropriate method and procedure for replication of data. The Baan Exchange Tool, or a replication mechanism provided with the database, may be used.

Central Purchase Contracts with Decentral Purchase Orders

Background of the Change

Prior to this change, the standard functionality assumed that purchase contracts would not be centrally managed. The functionality did not provide an option for storing and analyzing contract history for multiple logistic companies from a single company.

Many of Baan's customers do, however, manage purchase contracts centrally. The entry, maintenance, and analysis of the contracts is done by one department, while the contract terms and pricing apply to the entire organization, and therefore to all logistic companies in the organization.

For this reason Baan has changed the standard software so that central contract management is possible.

Description of the Change

The logistic company number is now stored in the purchase contract history, with this information being available in the sessions accessing contract history.

If contracts are to be centrally managed, the contract tables may be logically linked. If the structure consists of more than one server, replication of data across servers, not logical linking, is advised. This method is preferred because network downtime on the contract server would cause contract data to be inaccessible to the logistic companies.

How this functionality is used depends on the initial setup of the logistic company tables.

Restrictions

If contracts are created at both the central company and other logistics companies, ranges of contract numbers should be used.

From the central contracts company it is not possible to define delivery schedules or binding quantities per company.

Multiserver Customer and Supplier Files

Background of the Change

Prior to this change, standard functionality allowed central management of the customer and supplier tables through logically linking the tables.

When the structure consists of more than one server, Baan advises against logically linking tables across servers because network downtime on the central server would cause downtime for processing in finance and logistic companies on other servers.

The replication of data can usually be used to keep master data consistent across companies and servers. However, the fields order and invoice balances, and the five turnover values are dynamic, and may be updated by any company.

For this reason Baan has changed the standard software so that, while not logically linking the tables across servers, the values of these fields are updated for all companies on all servers, and are available to all companies on all servers.

Description of the Change

Two new tables have been added to manage the balance and turnover fields from the customer and supplier files. These tables may be logically linked within a server, where they will store the data by company number. The data in the tables must then be replicated across servers, so all servers have complete data for all customers and suppliers in all companies.

All existing sessions that read these fields have been changed to read from the new tables (e.g. credit checking).

3 Finance

3.1 Accounts Payable (ACP)

Central purchase invoice matching & decentral purchase order processing

In earlier versions the purchase tables of companies within one group had to be linked physically. This could cause the following problems:

- If, for some reason or other, the server goes down and one central purchase order file is used, a large part of the companies can not work anymore.
- The shared purchase tables became very large which can result in disk space and performance problems.
- If several companies made use of the same order number series, the six positions available for the order number in some cases were not enough to produce unique order numbers.

In BAAN IV b, all logistic companies belonging to the same group can get their own purchase tables. This solves the problems mentioned previously.

Authorization procedure for purchase invoices

The authorization procedure which was already available for preregistered invoices is now also implemented for registered invoices. This makes it possible to link an authorization scheme to a purchase invoice. After the invoice has been authorized by all persons and departments defined in the authorization scheme, it can be approved.

Document tracking

BAAN IV b contains a report for printing all postings that have been made for an invoice. The report shows all registration, approval, and correction postings for a particular invoice. After the invoice has been authorized by all persons and departments defined in the authorization scheme, it can be approved.

Parent-child structures for suppliers

Parent-child structures can be created by means of the existing factoring company functionality. The factor company is used as the supplier's parent company. It is possible to print open entry reports including data of related suppliers.

3.2 Cash Management (CMG)

Receipts for multiple companies

In BAAN IV b the process of assigning unallocated receipts in a multisite situation has been improved. In earlier versions an unallocated or advance receipt for invoices from several companies could not be assigned in one step. The received amount first had to be distributed over the companies, after which the amounts could be assigned to the invoices. In BAAN IV b the unallocated and advance receipts can directly be assigned to invoices of several companies.

Improved customer statistics

In BAAN IV b, the customer statistics (customer rating based on his payment record) have been improved. It is now possible to update the statistics for a range of customers. As it is no longer necessary to do a complete run of this program, the user can get up-to-date information on a particular customer within seconds.

New information in the statistics includes: the average days overdue, the credit limit, and the percentage of the credit limit used.

Improved cash forecast

In the Cash Management (CMG) module, a cash forecast can be generated. The purpose of this forecast is to provide information about the expected liquidity position for any currency on any date in the future. The cash forecast includes information from both the financial and the logistical applications: sales invoices, purchase invoices, sales orders, purchase orders, sales quotations, project orders, standing orders, and budgets.

Several inconsistencies in the cash forecast have been eliminated in BAAN IV b.

3.3 General Ledger (GLD)

Ledger and dimension history

In BAAN IV b more options are available for inquiries into the ledger and dimension history. It is possible to view transactions in the history for a range of periods. In addition, it is now easier to browse through the history information.

Column balance

In addition to the trial balance reports a new report is available to print debit/credit columns.

Multisite structure

Some changes have been made regarding intercompany and intergroup relations:

- Transactions between two or more groups of companies are temporarily recorded in the base company. These transactions become non-finalized transactions after running the Create Intergroup Transactions session. In earlier versions it was not possible to change the base company once it had been entered by the user. Now the base company can be changed if necessary, for example if the company structure must be changed at a later implementation stage.
- A report has been added to print intercompany relations for a selected range of intercompany relations.
- A report has been added to print intergroup relations for a selected range of intergroup relations.
- A report has been added to print a selected range of (non-) posted intergroup transactions.

Performance optimization

In BAAN IV b, the performance of many sessions has been improved. The most significant improvements are achieved in the area of posting integration transactions and the finalization of batches.

4 Manufacturing

4.1 Performance improvements

BAAN software opts for an optimal support of business procedures, which cannot be reached without the best performance. To ensure a high-level performance for various critical sessions, new features are available in BAAN IV b.

From now on programs can be run parallel to each other which means that from a program running in any bshell another program can be started in a separate bshell. This enables BAAN to spread workload across the available processor capacity.

Another feature to run programs faster than ever is loading various tables in cache memory to realize a very fast data access.

Several sessions have been improved with these two kinds of performance functionality.

Splitting of MRP & CRP parameters

A major change is done in the Capacity Requirements Planning (CRP) calculation principles. In former BAAN versions the planning horizon for Material Requirement Planning (MRP) and CRP could not be specified separately. Now changes were made so that MRP-CRP parameters can be split. A separate planning horizon can be specified for CRP. This useful and practical change will substantially reduce the MRP runtime to approximately 50% of the previous runtime.

The sessions affected in this case are Generate Planned MRP Orders (timrp1210m000) and Plan Production Orders (ticrp0201m000).

Lead time offsets

In addition to the split up of the MRP and CRP horizon BOM-items can have a Lead Time Offset (LTO). This means that if a CRP horizon has been specified and the requirement date is after the date determined by the CRP horizon, the start date will be calculated with the LTO data, kept per BOM component. So the start date will in such cases not be determined from the corresponding routing, which optimizes the run, since data so far ahead do not have to be that accurate.

Net change MRP

To simplify MRP net change runs the net change option will no longer be set for all components of one BOM but only for the component affected by a modification. This reduces the impact of a BOM change considerably.

These three topics all will raise BAAN to a higher performance level.

Net change MPS

A great new functionality of BAAN IV b is the possibility to generate Master Production Schedules (MPS) based on net changes similar to net change MRP. The Net Change MPS option can be set to Yes when a modification has occured which would cause an adjustment to the schedule.

So from now on it is possible to choose whether a regenerative or a net change run should be done or not.

5 Enterprise Modeler

5.1 Enterprise Modeler (BRG)

- A version may be protected by setting authorizations. Users can be authorized for all versions or for specific versions. The setting of authorizations is protected by a password. Versions without specific authorizations are fully accessible by every user.
- It is now possible to save utilities by version.
- Employees and Roles by Employee are now stored by Project Model.
- The number of positions for the employee code has been increased from 6 to 8.
- Parameters can be retrieved by their descriptions. Only a part of the description will suffice to find the parameter.
- Templates for the different text types can now be defined by version. An example of a text type is Business Function Text in the repository.
- A Petri-Net syntax check can be run on a number of Business Processes. Loops, Deadlocks, and Loose States can be traced in this way.
- An Activity Category may be linked to an activity.
- A Start State and an End State of a process can explicitly be determined.
- A File Browser has been created for importing an Enterprise Modeler version dump.
- The help pulldown menus of the editors now have the appearance of MS Windows applications. Help information on an activity or business function can be displayed by clicking on a help topic and selecting a component. The user will then be asked if the Common (repository) or Specific (model) help must be started.
- When leaving the business function Project Model, the user is asked whether or not to apply wizards. If all wizards are filled correctly, the user is also asked whether the parameters must be saved to the database. If the Model is changed (by adding or deleting business functions), the user is asked to perform consistency checks, transform BF Models into Business Process Models, set parameters based on rules, set static conditions based on rules, and save parameters to the database.
- An extra description can be defined by version and reference model. This description is displayed and printed at the bottom of the diagrams. The extra description of the version is displayed in the repository, and the extra description of the model in the model diagrams.

5.2 Wizards

- A start wizard is used to configure/model the BAAN application.
- In the business function Project Model user-friendly wizards are used to set company-specific parameters.
- Wizards can be changed/built into the repository and linked to business functions.

6 Transportation

6.1 Distribution Requirements Planning (DRP)

Containers in DRP

In BAAN IV a, containerized items were introduced in almost every relevant module, except DRP. With the introduction of BAAN IV b, containerized items are also fully supported in all business objects of DRP.

In the bill of distribution a hierarchical structure is implemented. The system first consults the bill of distribution on item/container level. If data is not available at this level, the item level will be consulted, and so on.

7 Constraint Planning

7.1 Intelligent Resource Planning (IRP)

The main change in BAAN IV b is the support for hybrid production situations with MPS/MRP for both non-Process and Process items, based on multilevel BOMs and formula structures. This functionality is designed for situations where Process and non-Process items are produced together.

BAAN IV b contains IRP, which provides most of the MPS functionality required for the Process package.

IRP introduces constraint planning techniques on MPS level. The BAAN IV b release of IRP combines constraint cyclic planning and traditional infinite MPS planning (with coproducts). Another important functionality includes extensive Available-To-Promise (ATP) techniques that can also be used to accept customer orders and determine delivery dates based on the availability of capacity and components. ATP can be checked and updated on item as well as on product family level.

IRP also supports central and decentral supply chain planning, both on item level as well as on product family level. Supply chain planning takes into account the constrained availability of capacity and/or components of supplying sites by using the advanced ATP techniques of IRP.

IRP supports MPS for containerized items. There are various possibilities to use the MPS planning for containerized items. In case of non-containerized items this distinction will not be made. See the following table:

	Main item MPS	**Main item not MPS**
End item MPS	Combined MPS	Containerized MPS
End item not MPS	Aggregate MPS	Not considered by MPS

An end item is a container, and a main item is a container-related item.

Containerized MPS:
In this case all end items are MPS items (it is not possible to have only one end MPS item), while the main items are not planned by MPS. So MPS planning and forecasting are only performed for the containers. The planning of the main item is less relevant.

Aggregate MPS:
In this situation, the MPS planning is only done for the main items.It is not possible to use MPS planning for containerized items (aggregate data).

Combined MPS:
In this scenario, the main item is handled as a product family, and is planned by MPS. The containers (end items) are also planned by MPS. So the MPS has multiple levels. Using this scenario, short-term planning and ATP checks can be performed at the container level. Long term planning (after the container specification time fence) can be done at the main item level (family level).

The following MPS functionality is supported by Manufacturing/Process rather than by IRP:

- MPS planning BOMs are applicable to containers.
- The MPS bill of critical materials may include both BOM and formula data. For example, an MPS bill of critical materials can contain lines not only for critical materials, but also for critical co- and by-products.
- Bills of material are **not** applicable to containers, only to non-containerized process items. Formula and Process routing information is considered.
- Interplant supplying sites are applicable to containers.
- Orders are generated by Manufacturing/Process, rather than by IRP. If IRP is used, MPS production orders, MPS production batches, MPS interplant orders, and MPS purchase orders are generated by MRP and stored as planned MPS orders. These are applicable to containers.
- Scheduled production batches derived from the MPS are generated by MRP and stored as planned production batches.

MRP considers the requirements for both Process items and non-Process items and uses them in planning PMG batches and SFC orders. In this way hybrid planning structures are supported for situations where both Process items (PMG production batches) and non-Process items (SFC orders) are used in production processes.

8 Tools

8.1 Baan Windows (BW) configuration

The configuration and and set-up of BW and its environment has been redesigned.

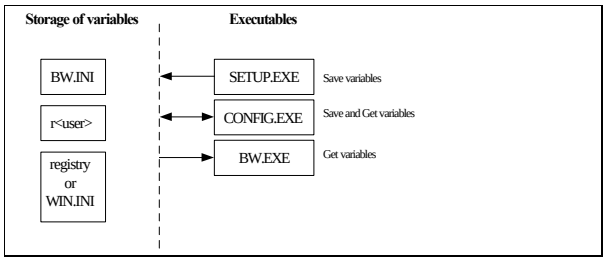


Figure 8-1 The old architecture

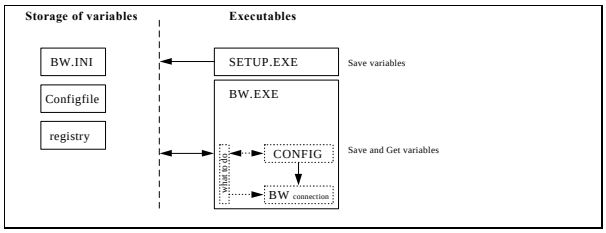


Figure 8-2 The new architecture

BW can directly connect to the BAAN system or start the configuration tool. Functionality has been added to the configuration tool which makes it more flexible to the end user.

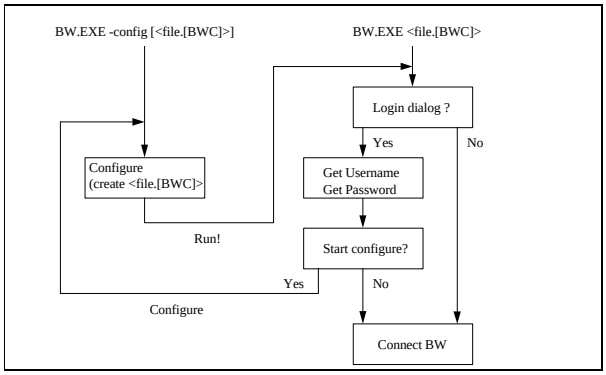


Figure 8-3 Flow of Start-up and Configuration BW

The configuration of BW can be started by running BW with the option *-config* from the login dialog or during initial setup.

BW can be started with *BW <file.[BWC]>. BWC* stands for BAAN Windows Configuration file. The configuration file can be activated by an icon or by the File Manager. Associate *.BWC* files with *BW.EXE*.

In the old situation all information such as defaults and configurations, etc. was stored in the *BW.INI* file. Due to the inheritance of BX, running under UNIX, there were r<user> files. These files have now been removed.

This means that algorithms have changed. We have now introduced configuration files. The *.BWC* configuration files hold all information needed to connect BW for only one connection to a server. These *<config.[BWC]>* files can be started directly by BW.

Password security

In the old situation, when creating a new configuration you needed to add the (encrypted) password to the *BW.INI* file. In the new *<config>.BWC* file it is still possible to add the password in the configuration file, but it is not necessary. The password field can be left empty. Before BW starts, the Login Dialog Box will pop up, in which user name and password can be specified at runtime. The password is not stored in a file.

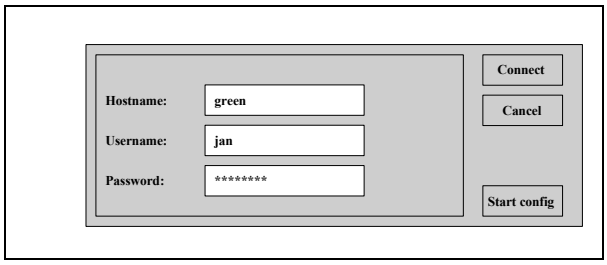


Figure 8-4 Popup of the Login Dialog Box

Figure 4 shows the Login Dialog Box, in which the user must enter a password. The user name can be changed. The Connect button activates the BW connection; the Start Config button activates the configuration tool.

8.2 Performance improvements

General

BAAN IV a has made a far greater impact on system resources than was expected during its development. To improve performance the tools have been optimized in several areas.

Since the increase in load of BAAN IVb is in the application layer, due to the GUI, it is recommended to use a client/server solution. The client hardware (application server) can be based on Windows NT and can be implemented in two different ways. One way is to implement an Windows NT application server, serving a group of users. The other way is to have every user running Windows NT on his desktop as an application server ('fat client'). Baan Development will test over the next couple of months what the requirements are to implement the latter. The server hardware (database server) could be either UNIX or Windows NT based, depending on the environment.

The improvements on the interactive processing are achieved by moving scratch tables to memory, removing the skip.io command in many sessions and changing SQL-statement to avoid unnecessary database I/O. This has reduced the incremental load of BAAN IVb. The decrease of the maximum number of users on a particular system with B**AAN IVb compared to TRITON 3.1b is now 20%** instead of 30%. This is observed in the host-mode with the Oracle database and TTY-interface.

Batch processing

Also batch processing is in many sessions optimized. The optimizations are achieved by reducing database Input/Output (I/O) and through parallel processing. The first way of optimization reduces the overall load of a batch process. The second way reduces the runtimes of batches by utilizing multiple CPU's in a system. The achieved improvements of the runtimes from batches is in between the factor 2 and 10. This depends on the type of batch session and the underlying data structure.

MS-Windows GUI

The MS Windows GUI makes higher demands on system requirements and resources. BAAN IV b includes a number of performance improvements. One example is the handling of menu bars and toolbars. As these are almost identical in all sessions, templates have been introduced to reduce network traffic. The reduction of network traffic is achieved by specifying the changes in the template instead of sending the complete menu bar or toolbar information. This performance measures does not affect the look of BW in any way.

8.3 Pre-install checks

A new tool called *preinst* has been developed. This tool checks, during an installation, whether several conditions are met. It does not update or change any parameters and/or environments. It only gives an advice. The tool is used before an installation or an upgrade. The objective is to prevent installations being aborted or having to be repeated. The tool may also be used to check operational BAAN software environments.

The main checks conducted by *preinst* are:

- Kernel parameters
- Database parameters versus kernel parameters
- Checking single/multi-byte database
- Using raw devices or file systems in case of a RDBMS
- Disk space and available space in table spaces
- Maximum file size
- BAAN users
- TCP/IP configuration
- Environment variables
- Runtime environment versus data dictionary
- Possible options for faster installation
- Install.data
- Checking tables
- Customization VRC
- Existence of fixes
- Client/server installation

The tool uses a configuration file which can be adjusted to new operating systems, new versions of BAAN software, etc. The tool uses this configuration file to perform the checks. The output of the checks are written to a log file, which may be used to solve problems, but which is also used for dynamic kernel tuning (if the operating system allows such tuning to take place).

8.4 Application function server

Most businesses today use a heterogeneous mix of applications. As these applications have to communicate with each other, each application must provide an Application Program Interface (API).

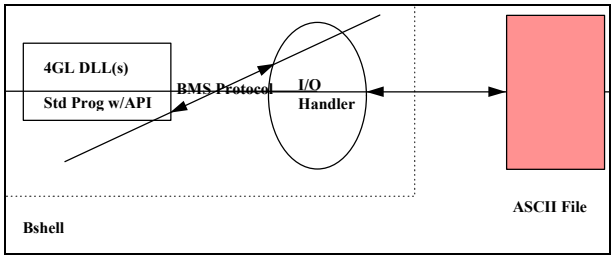
The BAAN family of applications is no exception. Thus far, the two major ways of interfacing with BAAN, the Exchange Tool and Distributed Data Collection (DDC), have failed to meet all of the following requirements for a BAAN API:

- Absence of code duplication
- Direct access to BAAN business logic
- No requirements for customer access to or use of BAAN application source code
- Easy transition from one version of BAAN to another (next) version

The API provides external (non-BAAN) applications access to the same business transactions that users have when using BAAN IV b through one of the standard user interfaces. By using the API, tools such as DDC and Exchange can meet all the requirements mentioned previously.

To this end the standard program has been extended. The standard program with API provides a standard set of primitives that external applications can use to perform actions against the BAAN database using BAAN business logic. This set of primitives corresponds to the current “choice commands” available to users of the BAAN applications. Since this version of the standard program is derived from the original, the environment remains the same. The primitives are accessed by means of the BAAN Message System (BMS) protocol available in the bshell.

A possible usage of this product is shown below:



In this diagram, a 3GL I/O Handler program reads an ASCII file and updates the BAAN database by using the primitives provided by the standard program with API. It is possible to replace the I/O Handler with an Exchange Object or a DDC function server.

The users of this product are application programmers constructing interfaces between BAAN and other (third-party) programs.

Due to the nature of the primitives provided, the standard program maintains a state (or set of values for its variables). This restricts the architecture so that one instance of the standard program can only talk to a single instance of the I/O Handler (or another process like that).

This product will be superseded by the new interface methodologies being developed in BAAN V and BAAN VI. The functions provided in this interface will be supported in these new releases. The functions will become available in DLLs in later versions.

8.5 Code instrumentation

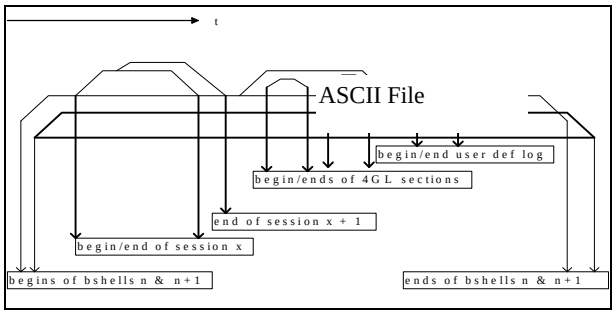
Code instrumentation as a built-in facility of BAAN IV b is meant to obtain insight in the processing time consumed by the standard program and the 4GL sections it handles. It is designed for workbench developers without tools authorizations to test the performance of their newly developed software.

In order to gain time performance insight, the beginning and ends of sessions, 4GL sections, and other user-defined program parts are logged. The logging itself must be distinguished from the storing and retrieval of the time flow information.

The following picture shows the logging of events on different levels: for sessions, 4GL sections, and user-defined log points. The solid arrows stand for the log points.

The Code instrumentation addition to BAAN IV meets the following requirements:

- Logging time flow data
  Time flow data is logged to gain insight into the performance of BAAN IV sessions by placing time logs at carefully chosen source code points.

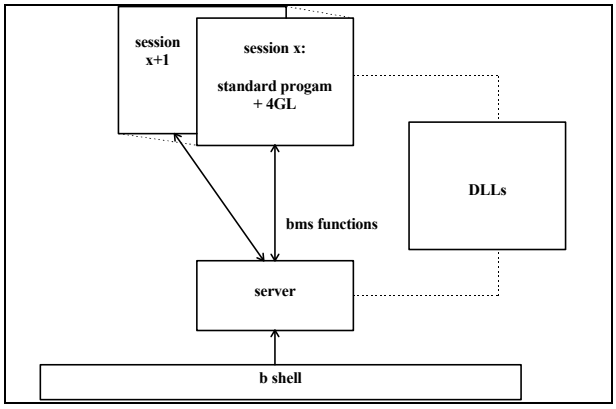


- Levelled logging
  The points for placing time logs can be divided into different levels.

level 0	start/end of 4GL session
level 1	start/end of every 4GL section (e.g. before.input, after.choice.update, etc)
other levels	start/end coded by the user in the 4GL scripts

- Optional logging
  Running BAAN IV with or without time logging, depending on the user's choice.

The following picture gives an impression of the BAAN IV environment in which the logging mechanism is integrated. The picture shows one bshell, although the user can activate multiple bshells. Each bshell starts its own server. It is the server that handles the multi-tasking of different sessions. Both sessions (standard program and the 4GL scripts) and the server are able to execute DLL functions.



Log points will be inserted in the standard program and at any place in 4GL scripts. These log points (functions) make use of the available data at the moment of logging, for example a session name. For identification reasons two numbers need to be added:

- The bshell number involved
- A log event number

The server needs to add these indexing numbers for two reasons:

1. Each server adds the bshell number of the bshell by which it is started; there is no need to send this bshell number to every session that is started.
2. The server can ensure a unique, incrementing event number.

Considering the logging events, the important information is:

- Date
- Time
- User name (additional, not necessary)
- Level
- Session code
- Section name (for level 1 or higher)
- Start/end
- Comment (optional)

8.6 SQL processor enhancements

Several enhancements are implemented in order to improve the performance and stability of SQL processing. These enhancements do not affect the functionality of the database driver, they merely enhance the internal functioning of the SQL processing.

8.7 BBASE enhancements

Various enhancements are added to BBASE, especially in the areas of data consistency and recovery. As with the SQL processor enhancements, the functionality of the driver is not affected, only the internal functioning of the driver is improved.

8.8 Oracle driver with distributed option

The Oracle driver has been extended with a feature that enables the driver to access data in distributed databases using Oracle's Distributed and SQL*Net capabilities (database links). This allows distribution of application tables across different Oracle instances on different systems, while transaction consistency is enforced by Oracle's two-phase commit protocol.

The distributed option, when enabled through an environment variable or resource, requires a new format of the Baan configuration files (ora_users, ora_groups) and even requires additional configuration files. Therefore, this methods must be considered a completely new setup, which also touches security issues, like users, groups, and roles. The option is available for the combi driver, which means for both the level 1 and level 2 driver.

8.9 DB2/6000 database driver

DB/2 is a popular database from IBM that runs on MVS, AS400, and RS/6000. BAAN IV b supports DB2/6000 running on the RS/6000 under AIX.

9 Utilities

9.1 Distributed Data Collection (DDC)

DDC was originally released as a GLO (Global Localization for North America) enhancement in the 3.1b release. With the release of BAAN IV b, DDC is included in the standard product while being imported at the same time.

DDC lets users enter data into Baan IV b in real time by using devices other than traditional data entry terminals as well as letting users use foreign control systems. A number of different bar coding/keypad devices can be used to enter data. These devices can be handheld batch or RF controlled, as well as direct connect entry terminals. The DDC system currently supports bar code wands, badge swipe readers, laser scanners, CCDs and keypad entry.

DDC is independent of the bar code hardware selected by the customer. However, the hardware vendor must supply a software front end that controls their devices in order to communicate with the BAAN IV b interface. The hardware suppliers, EPIC and Intermec provided the original interfaces to DDC for 3.1b.

The DDC communication functions with the hardware will vary by vendor.

9.2 DDC setup

The exact menu of transactions available for a particular factory data collection device is controlled by DDC. Menus are defined in DDC and downloaded to the front end to control the data collection devices. The flow of the transaction prompts are also controlled by DDC. The transactions are defined in DDC and are also downloaded to the front end.

DDC supports multiple methods of data entry validation. The three methods supported include:

- Field validation within BAAN IV b
- Field validation by front end
- Transaction validation within BAAN IV b

Attributes can be assigned for each data field. These attributes are used by the front end for doing data validation locally before passing the request to BAAN IV b for full validation. This allows many errors to be caught at the data collection device level and decreases the amount of network traffic among the data collection devices as well as between the front end and BAAN IV b. The attributes include data typing, template matching, and prefixes (data identifier).

Each field in a transaction can contain a data identifier prefix to make sure the bar code is of the appropriate data type. The prefix is part of the bar code, but not part of the data item itself. It is stripped off of the bar code before presenting the rest of the data field for validation (if required). ANSI FACT-1 data identifier standards are used for prefix support.

9.3 Communication

Each transaction can be configured from the front end with the desired method of communication with the DDC module. On-line, offline and on/offline communication methods are all provided.

Prior to the BAAN IV b release, the user had to choose one of the following download options when starting the DDC server:

- Menus only
- Functions only
- Neither menus nor functions
- Both menus and functions

As of the BAAN IV b release, the user now only has the options to start the DDC server with a full download of both menus and functions or no download. If No Download is specified, the devices simply come online with their previous menu structures. If Full Download is specified, the new downloaded menu structure will replace the old menu structure on the devices. This was done to make programming for the front end vendors less complex.

9.4 Error processing and transaction logging

Transactions arriving to DDC in offline mode that contain errors are not returned to the front end for correction because the DDC module assumes that the user is no longer available at the data entry device for error correction. The incorrect transaction is logged to an error log for later correction and processing.

The user is provided with a transaction error correction screen in the DDC module to correct errors in this log. The corrected transaction will be immediately submitted to BAAN IV b for processing. The user has the ability to report and manage the error log.

Since online transactions that fail validation are immediately returned to the front end and to the data collection device for direct correction, no online transactions are written to the error log.

All transactions that successfully update BAAN IV b are written to a DDC transaction log. Capability to report and manage the DDC transaction log is available.

9.5 Function servers

Function Servers are the BAAN sessions that:

- Perform a transaction
- Validate a field against a table
- Download data to the front for local validation

The following BAAN functions were originally released in 3.1b GLO and are now available for BAAN IV b:

- Labor Hours Accounting (HRA) (Manufacturing)
- Deliveries (SLS)
- Receipts (PUR)
- Cycle Counting

The following BAAN functions are new in BAAN IV b:

- Deliveries for ILC (SLS)
- Receipts for ILC (PUR)
- Generate and Release Inbound Movements (ILC)
- Generate and Release Outbound Movements (ILC)
- Enter Inventory Transfers (ILC)
- Cycle Counting (ILC)

9.6 Function server generation

A new tool for generating function servers is being introduced in BAAN IV b. This tool significantly cuts down the development time in creating new function servers.

Creating a function server is simply a programming process that converts a standard BAAN session into the function server. This typically includes writing 4GL code to convert the basic session function into the function server as well as making additional changes to handle error checking and handling.

The function server generator cuts down the development time by creating most of the code for a new function server. It uses the original script of the session that is being converted to create the actual function server. Depending upon the complexity of the original session, a generated function server script is anywhere from 75% to 95% ready to run.

9.7 Function server testing

A function server testing process is introduced in BAAN IV b to let the user test a new function server without requiring the use of actual collection devices. This function is useful in testing and debugging new function servers without disrupting the DDC system already in use on the shop floor.

9.8 Miscellaneous

Utilities are available for maintaining DDC message queues and for general performance tuning.

10 Available Documentation

10.1 Product Information

New Functions in BAAN IVb GLO	P3001A US
Sizing Guide TRITON 3.1/TRITON Tools 6.1	P3002A US
Sizing Guide BAAN IV	P3004A US

10.2 User Documentation

Installation Guides

Installation Guide for BAAN IV on UNIX	U7016A US
Installation Guide for BAAN IV on ORACLE	U7028B US
Installation Guide for BAAN IV on INFORMIX	U7029A US
BAAN IV Installation Guide for DB2	U7030A US

Conversion Guides

Conversion Guide TRITON 3.0 /3.1 to BAAN IV	U7025A US

User Manuals

Common User Manual 1 (COM,MCS, Parameters)	U7007A US
Common User Manual 2 (EDI)	U7006A US
Constraint Planning (CLP,RMP, RPD,RPU)	U7002A US
Distribution User Manual 1 (CMS, RPL, SMI)	U7008A US
Distribution User Manual 2 (PUR, PST)	U7009A US
Distribution User Manual 3 (SLS, SST)	U7010A US
Finance User Manual 1 (ACP, ACR, CAL, FBS)	U7011A US
Finance User Manual 2 (CMG, GLD)	U7012A US
Manufact. 1 (ITM, HRA, BOM, ROU, CPR, RPT)	U7013A US
Manufact. User Manual 2 (MPS, MRP, CRP, SFC)	U7014A US
Manufacturing User Manual 4 (GRT, PCF, Param)	U7015A US
Process Reference Manual Keywords&Methods	U7004A US
Process User Manual (FRM, ROU, PMG, Param)	U7005A US
Tools Programmer's Manual	U7021A US

For User Manuals about other modules: see the documentation for BAAN IV a

Miscellaneous

BAAN IV b Rapiditas Performance Guide	M2002A US

10.3 Training Materials

Module Descriptions

Constraint Planning (Resource Planning Data)	T6013A US
Constraint Planning (Resource Plan Units)	T6014A US
Constraint Planning (Resource Master Planning)	T6015A US
Constraint Planning (Cyclic Planning)	T6016A US
Enterprise Modeler	T6004A US
Supply Chain Sales Schedule Control	7796 US
Supply Chain Purchase Schedule Control	7797 US
Supply Chain Self Billed Invoicing (SBI)	7798 US

For other Module Descriptions: see the documentation for BAAN IV a.