BaanIVc4 Engine User Guide

Data Conversion Adapter 3.4 Suite

Copyright © 2023 Infor

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Release: BaanIVc4 Engine

Publication date: April 24, 2023

Document code: U9149F US

Contents

About this guide **7**
Intended audience 7
Related documents 7
Contacting Infor 8

Chapter 1 Introduction **9**
Purpose 9
Scope 9
Definitions, acronyms, and abbreviations 9
Need for the Tool 12
Components 12
User Profile 12
Authorizations 12
Customizations 12
Test Conversion 13

Chapter 2 Conversion Scenario **15**
Single Machine Scenario 16
Two Server Scenario 16
With client-server connection 17
Without client-server connection 17
Scenario Selection 19

Chapter 3 Migration Procedure **21**
Migration Procedure Activities 21

Chapter 4 DCA Infor BaanIVc4 Engine sessions **27**
Conversion Setup 27
Parameters (bcgct0100m000) 28

Source Companies to be Migrated (bcgct9189m000) 33
DCA Generated Conversion DLL objects (bcgct0106m000) 34
History/Multiple dumping Process per Target Table (bcgct8571m000) 35
Data Selection range per Group per Target Table (bcdmp8572m000) 37
List of Target tables being converted (bcgct9188m000) 38
Related Tables of a M-Object (bcgct9585m000) 41
Validate Data Integrity (bccnv4233m000) 42
Conversion Codes (bcgct9190m000) 43
Values by Conversion Codes (bcgct9191m000) 43
Old Values from a table field (bcgct9291m100) 45
New Values using DLL function (bcgct9291m200) 45
Maintain Pre-Migration session (bcgct0103m000) 46
Import Start-up sessions from a Flat File (bcgct0104s000) 47
To debug the generated script (bccnvdebug) 48
Conversion Process 49
Dump Baan IV data (bccrun9087m000) 49
Attention points 51
Reset the conversion process 51
Number of dump engines 51
Data Files 52
Archive Data Dumping Run (bccnv5201m000) 52
Check Currently logged in users (Unix only) (bccnv5406s000) 53
Check Environment of present Company (bccnv6400m000) 53
Check Scheduled jobs (bccnv5404s000) 54
Print Dumping Status (bccnv9442m00) 55
Functional error Logs during dumping (bcrun9599m000) 55
Data dumping Runs History (bccnv5500m000) 56
Display History – Conversion Time log (bccnv5501m000) 56
Error Logs - Archive (bccnv5503m000) 57
Print History – Conversion Time Log (bccnv5401m000) 57
Reset Conversion Parameters (bccnv0200s000) 58
Generate List of Log files to analyze (bccnv5301m000) 58
Analyse log files (bccnv5300m000) 58
Conversion Simulation (Optional) 59
Simulate Data Conversion (bccnv8101m000) 60
Simulation Run Details (bccnv8502m000) 60
Display History of Simulation Run (bccnv8503m000) 61
Print History of Simulation (bccnv8403m000) 61

Chapter 5 Special Migration Features 63
To copy data across companies 63
To migrate Text Tables (tttxt001, ttxt002, and tttxt010) 68
Split mapping objects using History/multiple dumping process per table 69
Recoding 71
Unicode Correction 72
UTC conversion 73

Appendix A Validate Data Integrity 75
Purpose of the bdvalidate utility 75
Reporting Sessions 75
Bdbvalidate6.x Usage 75
Types of domain constraints 76
Filled constraint 76
Range expression constraint 77
Uppercase / lowercase constraint 77
Left / center / right alignment constraint 77
Legal / Illegal characters constraint 77
Magnitude of floats and doubles 78
Checked constraints 78

Appendix B DCA 3.x Differences 81
DCA3.2 81
Enhancements DCA32 81
Fixes DCA32 82
DCA3.3 82
Enhancements DCA33 82
Fixes DCA33 82

About this guide

This is the user guide for Data Conversion Adapter Infor BaanIVc4 Engine. This guide helps you to understand the process to migrate data from Infor BaanIVc4 to Infor LN environment. This is one of the components of data migration tooling set called Data Conversion Adapter (DCA).

Intended audience

This document is intended for InforBaanIVc4 experts with technical knowledge of the Baan software architecture, who wants to move Infor BaanIVc4 customer data to INFOR LN environment and use the Infor migration BaanIVc4 Engine.

Related documents

You can find the documents in the product documentation section of the Infor Xtreme Support portal, as described in "Contacting Infor" on page 8.

Data Conversion Adapter 3.4 - Infor LN Engine - User Guide	*U9130G US*
Data Conversion Adapter 3.4 - Infor LN Engine - Installation Guide	*U9131G US*
Data Conversion Adapter 3.4 - Infor B50c Engine – User Guide	*U9137F US*
Data Conversion Adapter 3.4 – Infor B50c Engine – Installation Guide	*U9138F US*
Data Conversion Adapter 3.4 - Infor BaanIVc4 Engine – User Guide	*U9149F US*
Data Conversion Adapter 3.4 – Infor BaanIVc4 Engine – Installation Guide	*U9150F US*
Data Conversion Adapter 3.4 – Legacy Engine – Installation Guide	*U9096G US*
Data Conversion Adapter 3.4 – Data Conversion Adapter Studio – User Guide	*U9093F US*
Data Conversion Adapter 3.4 – Data Conversion Adapter Studio – Installation Guide	*U9094F US*

Contacting Infor

If you have questions about Infor products, go to Infor Concierge at https://concierge.infor.com/ and create a support incident.

The latest documentation is available from docs.infor.com or from the Infor Support Portal. To access documentation on the Infor Support Portal, select **Search > Browse Documentation**. We recommend that you check this portal periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1 Introduction

This chapter provides information about the scope, purpose, system requirements, customization, and need for this tool. This document assists users of the DCA Migration Tool (from Infor BaanIVc4 to Infor LN) to understand the different steps involved and illustrates the required actions.

Purpose

The purpose of this document is to provide the detailed description of the data conversion process for Infor BaanIVc4 to Infor LN with respect to DCA Infor BaanIVc4 Engine.

Scope

This document covers the processes of the DCA BaanIVc4 Engine. The DCA BaanIVc4 Engine is primarily used to extract data from the Infor BaanIVc4 environment by running the conversion *dlls* generated by the DCA Studio.

Before running the DCA BaanIVc4 Engine, you must finish the pre-conversion activities, such as maintenance of default data, generation of some mapping schemes, and so on. This is done in sessions exclusively created as a part of the Logic pack. These sessions are not explained in this document. For the details of these sessions see the documentation of logic pack.

Definitions, acronyms, and abbreviations

This table shows a list of definitions and abbreviations used with the BaanIVc4 Installation and user guide:

Acronyms	Definitions
4c-<number>	This refers to the activity process number in the BaanIVc4 Engine

Acronyms	Definitions
BaanIVc4 Engine	Data Conversion Adapter 3.4 – Infor BaanIVc4 Engine. This is used to carry out the data conversion program generated by the DCA-Studio and extract the data from the Infor BaanIVc4 environment.
BC	Baan Conversion package. All the components of the conversion software are under the “bc” package.
BSE/$BSE	BSE refers to the Baan Shell Environment. By means of the BSE environment variable, a link is established to the directory in which all Baan programs and directories are located.
Conversion	In this document, the terms migration and conversion are synonymous. It stands for the migration of data across various release of ERP.
Data Conversion/ Data Migration	The conversion of all the existing company data to the newly installed ERP environment.
DCA	Data Conversion Adapter. It is a set of tools that assist you to migrate data from a legacy ERP or Baan ERP environment to an Infor LN environment.
DCA Logic Pack	A Logic Pack is the complete set of components required to handle a Conversion from a specific ERP Source version to an Infor LN release. The Logic Pack consist of these components: Rules Pack (DCA Studio Rules, Conversion Rule Objects). Pre-migration Components (Parameters, tables, sessions, and programs) to installed at the source environment. Post migration Components (Parameters, tables, sessions, and programs) to install at the target environment. For example, the Logic pack Infor BaanIVc4SP15 to Infor LN 10.5. This logic pack consists of the conversion components to Migrate data from BaanIVc4 Service Pack 15 to the Infor LN 10.5 release.
DCA Object Pack	The (DCA) Object Pack are the conversion rules programs as executables and can be installed at the source environment. They are used by the BaanIVc4 engine to extract the source date. You can create the object pack from the DCA studio. With the logic pack delivery, also a standard Object Pack is available.
DCA Rules Pack	The (DCA) Rules Pack consists of these components: DCA Studio Rules. DCA Object Pack. The DCA studio Rules can be imported into the DCA Studio for making customizations to the standard delivered Logic Pack.

Acronyms	Definitions
	The (DCA) Object Pack is used by the BaanIVc4 engine to extract the source data.
DCA Studio	Data Conversion Adapter 3.4 – Conversion Adapter Studio. A tool to create the Logic Pack Rules Pack. The Rules Pack is used by the DCA BaanIVc4 Engine to do the actual data extract of BaanIVc4 environment.
Environment	The location, which consists of the directories on a system used by the programs.
LN Engine	Data Conversion Adapter 3.4 – Infor LN Engine. This loads the data files created by the BaanIVc4 Engine to the Infor LN application tables.
LN-<number>	This refers to the activity process number in the LN Engine.
Load	The process by which you load data to your target application.
Logic Pack	The conversion rules for a specific point to point data migration written in the DCA Studio and the corresponding Pre-migration and Post-migration sessions. For example, Logic pack of Infor BaanIVc4 to Infor LN, contains a Rules Pack, that can be imported to DCA Studio, and, BC package dumps that can be imported to Infor BaanIVc4 Environment (Pre-Migration sessions) and Infor LN Environment (Post-Migration sessions)
M-Object (M-script)	This is a conversion script. On execution, this creates data files for several Infor LN application tables. A m-object is a group of target tables, and one table in this group is designated as the ‘base’ table and all other table as ‘related tables’
Recoding	Recoding means changing your master data codes. For example, you have created your address codes in InforBaanIVc4, in an unorganized manner, you like to put some standard in address codes, and then during data migration, this kind of cleaning process can also happen.
Software Conversion	The conversion and upgrade of general or in-house customized software components to a new version of the ERP.
VRC	The Version Release Customer (VRC) module in Baan Tools is used to manage various versions and releases of a package. These four types of VRCs are available: L: Localizations B: Line of business-specific software C: General customizations O: Customer’s own customizations

Need for the Tool

The tool enables the current BaanIVc4 users to migrate their Infor BaanIVc4 data to the Infor LN environment quickly and efficiently.

Components

Infor BaanIVc4 Engine is delivered as **BC** package, in modules **CNV**, **GCT**, and **RUN**.

User Profile

You must have good knowledge of Infor ERP Baan and these concepts:

- Package combinations, package VRCs and companies.
- The Baan data dictionary structure.
- Baan database definitions, such as physical/logical tables and table sharing concepts.
- The reconfiguration of tables.

Authorizations

You must have the super user authorizations and Baan Systems Administrator permissions to use this tool. Ensure that you have enough disk-space and write permissions to the directory of Engine files and directory of data files. These directories are specified in **Conversion Parameters (bcgct0100m000).** In case the directory is not present, then you must create this directory in both the BaanIVc4 Engine and LN Engine environments.

Customizations

You must use the standard rules pack to convert a standard Infor BaanIVc4 database (non-customized data model) to a standard Infor LN.

If the Infor BaanIVc4/Infor LN software contains customized table definitions, the standard conversion rules must be modified to fit the customer's specific configuration. To customize the standard rules pack you must use the DCA Studio tool.

Note: For a non-standard data model, you can only carry out the test conversion after the customizations are redefined in Infor LN, and the conversion program is adjusted to the Infor LN table definitions. User must check the derived-from structure for all possible types of non-standard software, that is, all types of localizations (L-VRCs), line of business-specific software (B-VRCs), general customizations (C-VRCs), and in-customer's own customizations (O-VRCs). If any of these VRCs are present, user must check whether the VRCs contain customized table definitions, and ensure the conversion rules are adjusted to these definitions, if necessary. You must not carry out a conversion before this issue is resolved.

Test Conversion

The test conversion simulates the final conversion with these objectives:

- **Tune the conversion environment to customizations (if any)**

  If the Infor BaanIVc4 version includes customized table definitions, you must modify the standard conversion rules pack to fit the customer's configuration. In this case, you can carry out the test conversion only after the customizations are redefined in Infor LN.

- **Provide information about the run time of the actual conversion**

  The plan for the final conversion is based on the run time of the test conversion. An estimate of the actual downtime can be made based on the run log information.

- **Provide certainty about the progress of the conversion run**

  The final conversion must take place without errors. In case any error occurs during the test conversion, the causes must be traced and cleared so that the test conversion can be repeated without errors before the final conversion.

- **Provide certainty about the results of the conversion run**

  The results of the test conversion are assessed based on the data and procedure tests. Because all functions and procedures in the Infor LN application can be used, based on the data of the user's own company, maximum certainty is achieved with regards to the continuity of business processes.

Chapter 2 Conversion Scenario

The data conversion process depends on the deployment environment of the customer system, such as installation of Infor BaanIVc4 and Infor LN on the same server or different servers. The conversion scenario is specified in the Conversion Parameters sessions and the Two-machine scenarios required some extra setup and configuration of directories. The various conversion scenarios are explained later.

This chapter describes the conversion scenarios in detail with its architecture. In this Chapter the source ERP environment refers to Infor BaanIVc4 and the target ERP environment refers to Infor LN.

Single Machine Scenario

In the single machine scenario both the Source ERP and the target ERP environments are installed on the same Server.

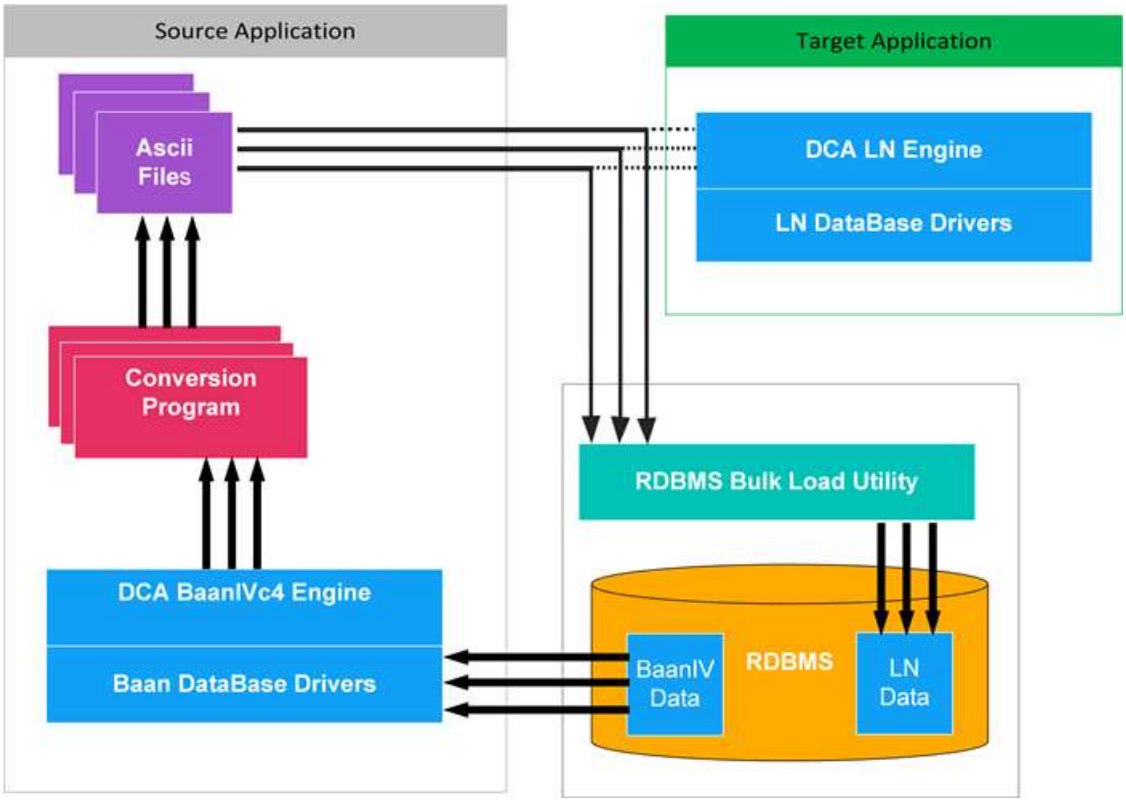

Figure **Error! No text of specified style in document.**-1 Single server scenario

Two Server Scenario

The source and target environment are installed on two separate machines. In this setup, these two cases are possible:

- Two server scenario with client-server connection
- Two server scenario without client-server connection

With client-server connection

On the target Environment, a Source environment FAT Client is installed. The DCA LN Engine uses this FAT Client to access the resource of actual source environment, to carry out the conversion programs.

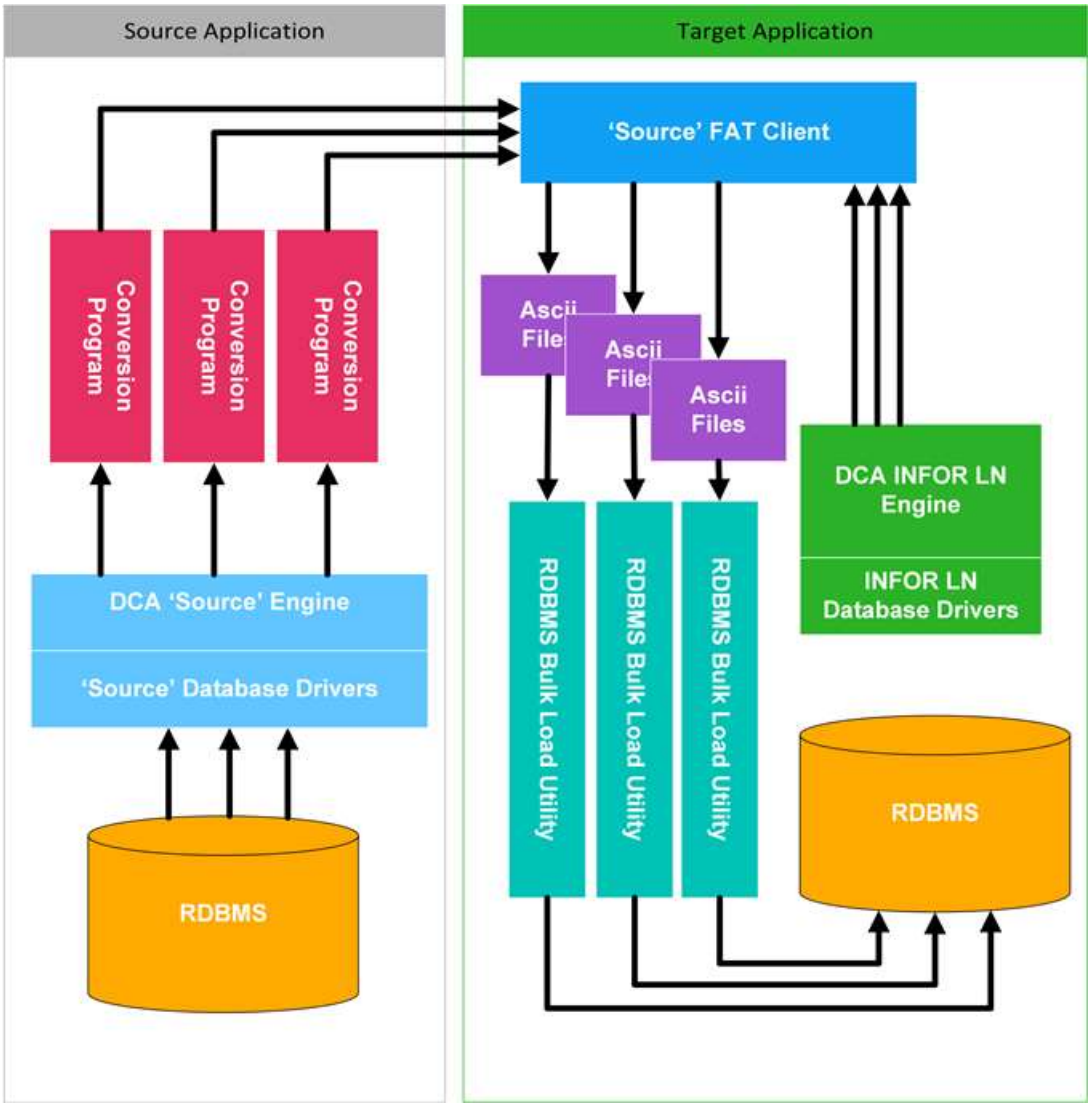

Figure **Error! No text of specified style in document.**-2 Two server with client

Without client-server connection

You can use this conversion scenario when no network connection exists between the Source environment server and the target environment server. You can also use this conversion scenario when the “Two server with client-server connection” configuration is not desired or is not possible.

The data dumping process starts in the DCA source environment Engine and creates the data files. After the data dumping is over, these files must be copied or transferred to the Target environment server. Start the DCA Target Engine to load these data files to the database.

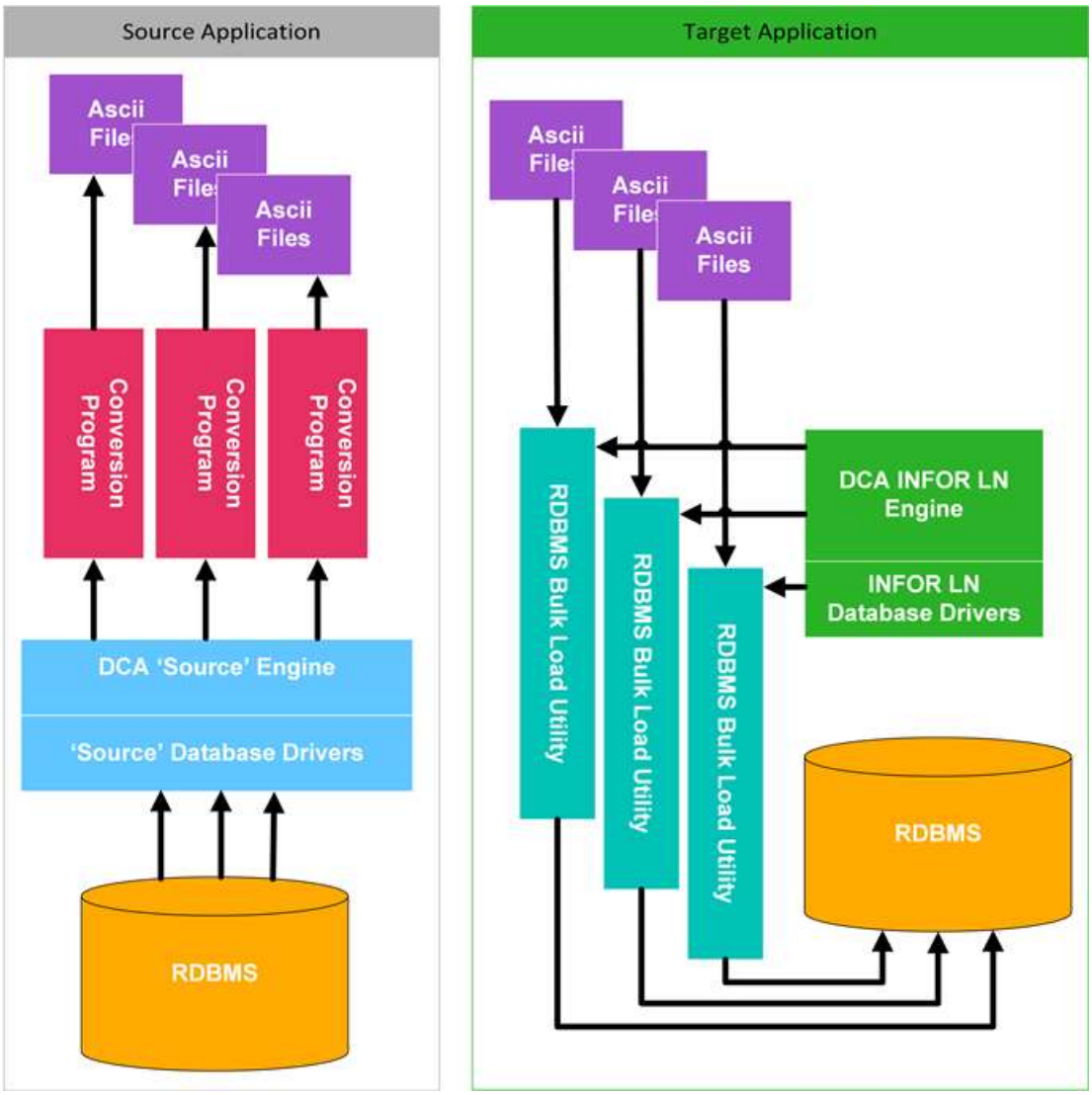

Figure **Error! No text of specified style in document.**-3 Two server without client

By default, in this method, you dump the data in the source environment and later load these files to your target environment. If you can create a shared drive between the two environments a parallel synchronized dumping and loading process of the data is possible. The migration user from source environment and target environment must have read/write permissions of all the files in the shared drive else the synchronized process is not possible.

Scenario Selection

The two conversion scenarios are referred to as the single-machine scenario and the two-machine scenario, respectively and the two-machine scenario has two cases: With or without a client-server connection.

Before starting the conversion, you must determine which of the two conversion scenarios best suits the specific situation at the migration customer. You must understand the impact of all the choices presented in the flowcharts, particularly the impact of the customized table definitions for a nonstandard data model (DM).

Pros and Cons of the Two Scenarios

Conversion Scenario	Advantages	Disadvantages
Single Machine	Dumping and loading can be performed in parallel. No network interference Simple Setup.	Because both source and target environments are on the same machine, the server capacity, such as CPU and disk space, can be a bottleneck.
Two Machine (With Client Server Connection)	Dumping and loading can be performed in parallel.	Increased network traffic. Setup involves additional activities, such as FAT client and remote-user connection, which can be complex.
Two Machine (Without Client Server Connection)	Performance advantage. Dumping and loading happen on two different servers. Complete server capacity can be utilized to perform either dumping or loading.	The dump (data) files are moved from the source server to the target server. By default, dumping and loading cannot be carried out in parallel unless a shared drive can be created.

If the source and the target environments are on two different servers, you can also use the single Machine scenario. If the source and the target server OS is also the same and a shared directory is in place, it is also possible to run the dump and load processes synchronized. But in this case, you must manually alter the Conversion parameters. In the LN engine set the “Source Host Name” parameter to the IP address of the source server and set the “Common” variable in the BWC file of the LN BWC to the shared engine directory.

Note: The installation process of both source environment and target environment migration software remains the same in all scenarios.

Chapter 3 Migration Procedure

This chapter explains the activity list and the detail process flow in the BaanIVc4 Engine and LN Engine. These are explained per data conversion scenario.

Migration Procedure Activities

The activity list provides an overview of the activities required for the conversion from Infor BaanIVc4 to Infor LN on a time line. The activities having the same number can be carried out simultaneously. If the activity carries out or maintains data in a session, the details for the BaanIVc4 Engine sessions of these sessions can be found in the following chapter. For the details on the LN session see the *Data Conversion Adapter 3.4 - Infor LN Engine - User Guide (U9130F US)*

This conversion process is applicable for a single company migration scenario. For example, you can migrate a BaanIVc4 company 100 to Infor LN Company 100. The process is also similar for a cross company migration. For example, from BaanIVc4 company 100 to Infor LN company 200.

The delivered migration Logic Pack also consists of a Data migration process checklist Excel file. In this checklist, all the data migration procedure activities are listed in sequence from installation of the engines and the logic packs to handling pre-migration sessions and post migration sessions.

Note: You can perform DEM model conversion before the Infor LN environment goes live.

Prerequisites
Check 1: If there are customizations and or localization present in the BaanIVc4/Infor LN Data models, load these customized data models to the DCA Studio and create a customized migration profile. Derive this customized profile from the standard migration profile you have received as a part of the Rules Pack. Identify the conversion objects to modify, and copy these objects to your customized profile. Complete the required changes (Use the data model comparison feature in Studio to identify the changes between standard data model and customized data model). After you have completed the changes, carry out a Conversion build. Conversion build compiles the customized Rules pack to the Object pack. For more information on how to work with DCA Studio, see the DCA Studio User Guide.
Check 2: Complete these installations:

Prerequisites
• Install BaanIVc4 Engine • Install BaanIVc4 Logic pack (pre-conversion parameters and mapping sessions) • Install Infor LN Engine • Install Infor LN Logic pack (post-migration sessions)

Check 3:

You must obtain a license for your migration project, and configure the license accordingly. To know more about licensing process, see BaanIVc4 Engine Installation Guide (U9150F US) and Infor LN Engine – Installation Guide (U9149F US)

Check 4:

You must decide on the data-conversion scenario you want to use. For more information on data conversion scenarios, see **Error! Reference source not found.**.

If the scenario is Two machines with Client-Server, then you must install the FAT client of Infor BaanIVc4 on the Infor LN Environment, and configure it to connect to the master BaanIVc4 Server, using the user, that is used to perform the conversion run.

If your scenario is Two machines without client-server, and you like to carry out data dumping and loading in parallel, complete these directory setups:

BaanIVc4 Engine and LN Engine are on Unix: In this case, you must mount one directory in the LN Machine to BaanIV machine preferably in the same name for easy maintenance (do not mount a directory in BaanIVc4 machine to LN machine). Under this directory you can create one directory, say “data” to store the data files and another directory, say “share” for both Engines to communicate. Ensure that the “data” directory has enough space to hold the temporary data files created.

BaanIVc4 Engine and LN Engine are on MS Windows: In this case, you must share a directory in the LN Machine. Under this directory you can create one directory, say “data” to store the data files and another directory, say “share” for both Engines to communicate. Ensure that the “data” directory has enough space to hold the temporary data files created.

BaanIVc4 Engine and LN Engine are on different operation systems: In this case if you can mount a directory in LN to BaanIVc4 machine, the setup is like case 1 above. Otherwise you cannot carry out a dump and load in parallel.

BaanIVc4-Engine Related Activities	LN-Engine Related Activities
4c.1. Maintain Parameters (bcgct0100m000).	Ln.1. Maintain Parameters (bcgct0100m000). You can start this session from the Dash board.
4c.2. Maintain Source Companies to be migrated (bcgct9189m000). **Note**: You can use the Generate option from the specific menu. This automatically inserts all the companies under your package combination. You can remove the unwanted companies.	Ln.2. Maintain Batch Codes (bcgct8188m000). You can start this session from the Dash board.

BaanIVc4-Engine Related Activities	LN-Engine Related Activities
4c.3. Get DCA Generated Conversion dll Object (bcgct0106m000). **Note**: You must use the Generate option from the Specific menu.	Ln.3. Maintain the Tables by Batches (bcgct8189m000). **Note**: This session can be started from the Specific menu of the Maintain Batch Code (bcgct8188m000) session.
4c.4. Generate target tables that can be migrated from the above list (bcgct9188m000). Note: You can use this option from the Specific menu of bcgct9188m000 session.	Ln.4. Generate OnpLoad Database (bcrun0207m000). This action is required only when your LN Application's database is Informix. You can start the session from the Specific menu of Batches (bcgct8188m000).
4c.5. Validate Data Integrity (bccnv4233m000). This process checks for the empty required fields in the source data, and domain constraints violations. This is an optional step. Validation of the data can also be performed after the data load at the target side.	
4c.6. Carry out Pre-conversion process activities of the Logic Pack	Ln.6. Carry out Pre-Conversion process activities of the logic pack.
You must complete these activities if you carry out dumping and loading in parallel. Otherwise, skip this block and complete the steps in section, Dump and Load later.	
	Ln.7. Synchronize with the source. In this process, you synchronize the information maintained in the LN Environment and Infor BaanIVc4 environment. **Note**: You can access this option from the Specific menu of the Dash board. Pre-conversion (With Client Server) → Synchronize with Source. You must also ensure that all activities including 4c.5 & Ln.3 are performed.
	Ln.8. Start the Conversion process (bcrun0200m000). Note: You can start this session from the Specific menu of the Dash board. Conversion → Dump and Load.
4c.9. Start Dumping (bcrun9087m000). **Note**: The action in Ln.8 starts the menu browser of Infor BaanIVc4 Engine. Start, the dumping process from this menu browser, Data Conversion Adapter – Infor BaanIVc4 Engine →	

BaanIVc4-Engine Related Activities	LN-Engine Related Activities
Conversion Process → Procedure → Dump Data (Create Data Files).	
	Ln.10. Perform the Post Conversion process activities.
Dump and Load later: You must carry out the activities from here if: First you must complete the data extraction (dumping) in the Infor BaanIVc4 environment, and after dumping, you can start loading the data files to Infor LN environment.	
	Ln.7. Export the target tables information. This action creates the file <Target Company Number>.tables.info, in directory <directory of Engine files>/<Target Company Number>. Directory of Engines files are maintained in LN Engine's Conversion Parameters. For example, if you have specified "Directory of Engine files " as, "/migration/share" and your current target company is "600", then the tables are exported to file, "/migration/share//600/600.tables.info". Note: You can access this option from the Specific menu of Dash Board (bcgcteasym000). Pre-conversion (Without Client Server) → Export Target Info. to Source. You must also ensure that all the activities including 4c.5 & Ln.3 are carried out.
4c.8. Import Target Info. from file. This file (<target company number>.tables.info) is generated by action Ln.7. You must copy this file from the LN environment to the Infor BaanIVc4 environment, to the location <Directory of Engine files>/<your source company>. The Directory of Engine files are maintained in your BaanIVc4 Engine's parameters. For example, if you have specified Directory of Engine files as "/migration/share" and the source company to be migrated is "600", then copy the file as: "/migration/share/600/600.tables.info". **Note**: You can access this option from the Specific menu of List of Target tables to be migrated (bcgct9188m000). You must also	

BaanIVc4-Engine Related Activities	LN-Engine Related Activities
ensure that all the activities including Ln.7 are carried out.	
4c.9. Export Source Information to file. This action creates the file <Source Company Number>.objects.info, in your <Directory for Engine files>/<Target Company Number>. For example, if you have specified the Directory for Engine files as ”/migration/share” and your current source company is “600”, then the tables are exported to file, “/migration/share/600/600.object.info” Note: You can access this session from the Specific menu of List of Target tables to be migrated (bcgct9188m000). You must also ensure that all the activities including 4c.8 are carried out.	
	Ln.10. Import Objects present information. This file (<target.company.number>.objects.info”) is generated by action 4c.9. You must copy this file from the Infor BaanIVc4 environment to your Infor LN environment, to the path <Directory of Engine files>/<your target company>. For example, if you have specified your Directory for Engine files as “/migration/share” and your target company is “600”, then copy the file as “/migration/share/600/600.objects.info”. **Note**: You can access this option from the Specific menu of Dash Board (bcgcteasym000). Pre-conversion (Without Client Server) → Import Source Info. To target.
4c.11. Dump Data (Create Data Files) (bcrun9087m000).	
	Ln.12. Transfer the data files generated by action 4c.11 to your Infor LN Server, in any directory and perform a ‘Load without dumping’ **Note**: You can access this option from the Specific menu of Dash Board (bcgcteasym000). Conversion → Only load. **Note**: In BaanIVc4 Engine, the created data files are organized in directories “<package><module>”, retain this directory structure during the file transfer. While loading the

BaanIVc4-Engine Related Activities	LN-Engine Related Activities
	loader searches for the files, as in the mentioned structure in path “<Directory of data files>/<target company number>”
	Ln.13. Carry out Post Conversion Sessions if any.

Chapter 4 DCA Infor BaanIVc4 Engine sessions

This chapter explains the sessions in the Infor BaanIVc4 Engine. You must start the BaanIVc4 Engine menu from the menu browser. Select **File → Run Program…,** and carry out the 'Program/Session' as "**mbcgctconvert**". This BaanIVc4 engine menu appears:

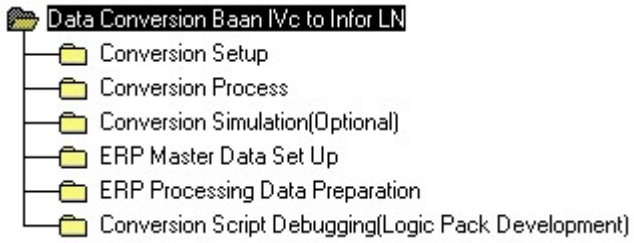

Figure **Error! No text of specified style in document.**-1 BaanIVc4 Engine menu

The next sections explain each of the sessions under this sub-menu. In this chapter no screenshots of the separate sessions are added but just the explanation of the fields in the sessions. To view the session, open the session via the specified menu or via run program.

Conversion Setup

Data maintained in the sessions under the Conversion setup are crucial to the migration process. This table shows the sessions under Conversion Setup and relate to setting up the BaanIVc4 Engine.

Session	Session Code
Parameters	bcgct0100m000
Source Companies to be migrated	bcgct9189m000
DCA Generated Conversion Dll Objects	bcgct0106m000
History/Multiple dumping Process per Target Table	bcgct8571m000
List of Target Tables to be Converted	bcgct9188m000
Validate Data Integrity	bccnv4233m000

Session	Session Code
Conversion Codes	bcgct9190m000
Maintain Pre-Migration Sessions	bcgct0103m000
Utilities	Submenu
Companies(*)	ttaad1100m000
Create Sequential Dump of Table(*)	ttaad4226m00
Create Table from Sequential Dump(*)	ttaad4227m000
Copy Company Data (*)	tccom0214m000
Create Tables (*)	ttaad4230m000
Delete Tables (*)	ttaad4231m000

(*) This is a standard BaanIVc4 session which is not described in detail in this document.

Parameters (bcgct0100m000)

The **Parameters ((bcgct0100m000)** session allows you to maintain the migration parameters. This session controls the overall data migration process. You can maintain only one set of parameters. The session comprises of two tabs: Engine details and options.

You must maintain the Conversion Parameters, in all the source companies, from which you want to migrate data.

This table shows the fields in the **Engine Details** tab of the session:

Fields	Description
Group	**General Details**
Source Company to be Migrated	The company for which you are maintaining conversion parameters. This is a non-editable field.
Conversion Scenario	Select the conversion scenario for the migration process. For more information on Conversion Scenarios, see **Error! Reference source not found. - Error! Reference source not found.** The available options are: Single Server: In this case, both the Infor BaanIVc4 and the Infor LN environments are deployed on the same machine. Two Servers with client-server architecture: In this case, a Baan client-server connection must be established between the LN Engine and BaanIVc4 Engine. This client server connection is achieved by installing an Infor BaanIVc4 FAT client in the LN Engine Server. The

Fields	Description
	LN Engine can connect to the BaanIVc4 Engine and carry out the data extraction process through this FAT client. Two Servers without client-server architecture: In this case the Infor LN and Infor BaanIVc4 are installed on two different machines. To dump and load data simultaneously you must mount a directory present in the Infor LN machine to Infor BaanIVc4 machine.
Directory of data files	Enter the directory to store the created data files. The data files are organized, as <package><module>, under this directory, during the conversion process. You must specify this directory, based on the conversion scenario. For example: Single Server: Specify a directory that is present on your Infor BaanIVc4 Server. Two machines without client server: Enter the directory, present on your Infor BaanIVc4 Server. For dump and load data simultaneously, you must mount a directory present in the LN machine on to BaanIVc4 machine, preferably of the same name. This example guides you what directory you must enter in this field, depending on Operating System, the name of the directories you can choose by your own. LN Server is UNIX → You have a directory “/migration”, in the LN Server; mount this directory as “/migration”, in the BaanIVc4 Server. Under this directory you can create two sub-directories “/migration/data” and “/migration/share”. Enter value “/migration/data” in the field. LN Server and BaanIVc4 Server are MS Windows → You have a folder “D:\migration” in the LN Server, share this folder, with permission “Everyone have full permission”. Under this folder you can create two sub-folders “D:\migration\data” and “D:\migration\share”. If the machine name of LN server is “Infor-LN”. You can enter the value “\\Infor-LN\migration\data” in the field. Two machines with client server: Enter the directory present on your Infor LN server. Because, the FAT client installation of Infor BaanIVc4 handles the data extraction process on the LN Server. The data files are created on the LN server itself. Note: Ensure that the conversion user has full permission for this directory and enough disk space is available to hold the data files.
Directory of Engine files	This directory is used by the conversion engines to share information and create temporary files. You must specify this directory, based on the conversion scenario. For example, Single Server: Specify a directory that is present on your Infor BaanIVc4 Server Two machines without client server: Enter the directory, present on your Infor BaanIVc4 Server. If you are doing a data dumping and

Fields	Description
	loading in parallel, you should mount a directory present in the LN machine to BaanIVc4 machine, preferably in the same name. This example guides you what directory you must enter in this field, depending on Operating System, the name of the directories you can choose by your own. LN Server is UNIX → You have a directory “/migration”, in the LN Server, mount this directory as “/migration”, in the BaanIVc4 Server. Under this directory you can create two sub-directories “/migration/data” and “/migration/share”. Enter value as “/migration/share” in the field. LN Server and BaanIVc4 Server are MS Windows → Say you have a folder “D:\migration” in the LN Server, share this folder, with permission “Everyone have full permission”. Under this folder you can create two sub-folders “D:\migration\data” and “D:\migration\share”. Say the machine name of LN server is “Infor-LN”. You can enter value “\\Infor-LN\migration\share” in the field. Two machines with client server: Enter the directory present on your Infor LN server. Because, the FAT client installation of Infor BaanIVc4 handles the data extraction process on the LN Server itself. **Note**: Ensure that the conversion user has full permission for this directory and enough disk space is available to hold the data files.
Path of Time zone files	Specify the path where the time zone directories: “zone info” and “zone rules” are located. Ensure that the application server run time and application components have the correct date and time value. In case of single machine scenario, you can point to the Target directory like /b61amig/bse/lib In case of a two-machine scenario, copy the zone rules and zone info files from the target $BSE to source $BSE. Set the BAAN_TZ_DIR to configure the time zone directory. **Note:** Enter the path of the parent directory of the “zone rule” directory without a trailing forward slash.
Your e-mail	Enter your e-mail address to receive dumping related error messages in case the migration scenario is a two-machine scenario.
Group	**Baan4 Windows Details**
Installed directory path (Local)	The Infor BaanIVc4 Windows installation directory in the local machine, from which you start the conversion process. You must enter this directory in the MS-DOS naming convention. For example, if you have installed Infor BaanIVc4 Windows in "C:\Program Files\Baan\Baan Windows"; the corresponding DOS naming convention can be "C:\Progra~1\Baan\BaanWI~1". **Note:** The DOS naming convention is machine dependent. To know the DOS name for a specific folder, start the command prompt, go to

Fields	Description
	the parent folder Type the command "dir. /x" to view the DOS naming convention for its sub-directories.
Bshell name	The Shell name of the BaanIVc4 environment. You can find the Bshell name in the environment configuration file (BWC file).
Protocol	The Protocol used by the BaanIVc4 windows to connect to the Infor BaanIVc4 environment. The available options are: Baan Login and Rexes.
Port number	The Port number used by InforBaanIVc4 windows to connect to the Infor BaanIVc4 environment.
User name	Specify the user name used to connect to the Infor BaanIVc4 environment. Ensure that this user is linked to the correct package combination, and has permission to retrieve data from required companies and full permission to all directory structures.
Password	Specify the password for the above specified migration user.
Group	**Target Environment Details**
Database	Select the database used by your Infor LN application. The database options are: • Oracle • Informix • DB independent • IBM DB2 • SQL server • EDB Postgres
Time-zone	Select the time zone of your Infor LN environment.
Copy to different Target company	Specify this field when your target company is different from the source company. This is referred to as ‘cross-company migration’ in this document. **Note**: If you are copying data to a different Infor LN company, you must map the source company to the respective target company, in the Maintain Codes for re-coding data (bcgct9190m000) session. Use the Code “COMP”, for this purpose. Also use the Re-coding utility in the DCA Studio, to correct the Rules Pack, because some data store the company number.

This table shows the fields of **Options** tab of the session:

Fields	Description
Group	**Performance Options**
Number of Data dumping processes	Specify the number of processes to be started to perform the data extraction process. A higher value improves the overall performance of the data conversion. However, examine the hardware profiles (Number of CPUs, RAM available, etc.) before you specify a higher value and take the conversion scenario and the dump and load process into account. **Note:** Take care of not using too much capacity of the system in case the capacity is shared with normal production (virtualization) **Caution:** The system creates individual shells for each process that it starts. Ensure that you have sufficient user licenses for each process that you start. For example, if you start 6 processes, then ensure that you have 6 free user licenses available before you start the conversion run.
Show record count while showing progress	Specify whether you want to see the number of records dumped. **Note**: If you set the field to Yes, the performance of data extraction reduces. You must also specify a higher value in the Refresh Interval field.
Refresh Interval	This field is enabled only when the Show record counts while showing progress? Field is set to 'Yes'. This value indicates the number of records the system dumps before the interface is refreshed to indicate the progress. **Note**: Specify a higher value for this field to improve the performance. For example, if you specify the value as 1000, each process after extracting 1000 records, updates the interface.
Store Dumping details	Select Yes to store these details to per target table: Duration of the data extraction process per target table. Number of records extracted per target table.
Group	**Miscellaneous**
Current Batch	This is a display field that provides information about the Batch you have synchronized from your LN Engine. This field is populated when you synchronize your batch details from LN Engine.
Master data conversion	This is a display field that indicates whether the current batch contains only the master data. This value is set in LN Engine and is populated during the synchronization.
Archive time log after conversion	Set the field to Yes, to archive the time log automatically after data conversion.

Fields	Description
	Note: if you set the field to No, you can use the Archive Data dumping run (bccnv5201m000) session to archive the time log after the data conversion.
Is Source environment Multibit?	Set the field to Yes, when the Infor BaanIVc4 environment is multi-byte.
Is L1 driver?	Select the type of the database driver available on the Baan database at the target side. Baan database has two types of drivers, L1 and L2.
Is Target UTC40 format?	Set the field to Yes, when the UTC date fields at the target side allow dates beyond “18-01-2038”. If the field is set to Yes the maximum value for the UTC date is set to “30-12-9999 12:00:00”. If the field is set to No the maximum migrated UTC date is “18-01-2038”. This value is used when the source date is beyond 2038 or the default maximum date is required.

Source Companies to be Migrated (bcgct9189m000)

This session allows you to maintain the list of source companies that must be migrated to Infor LN. In this session, you can maintain companies that are linked to your current package combination only.

Note: You must maintain this information in the company (your **current working company**) from which you start the data dumping process. You must not maintain this information in any other source companies you want to migrate.

The data maintained in this session is not cleared either after you perform dump only or dump and load. Therefore, prior to the conversion run, ensure that data is properly maintained in this session.

This table shows the fields and commands in this session:

Fields	Description
Source Companies	Enter the number of the company that you like to migrate. This field is enabled only when you add a new record. The corresponding company description is shown.
Commands	Description
Generate	Click “Generate” or select this option from the Special menu, to automatically add all the companies linked to your package combination.

DCA Generated Conversion DLL objects (bcgct0106m000)

This session allows you to notify the BaanIVc4 Engine about the available conversion script objects generated by the DCA Studio and or imported with the standard migration logic pack. You cannot maintain the data manually.

Select the **Generate** option, to automatically scan for the conversion objects in the vrc structure of the package combination. All the objects in the bc DLL module within the structure are made available. Therefore, ensure that you do not create any test scripts in the DLL module. After customizing the logic pack you must create a conversion build (Generate Scripts and Compile) from your DCA Studio before you run this session.

Note: You must maintain this information, in the source companies, from which you want to migrate data.

This table shows the fields and commands in this session:

Fields	Description
Package Combination	The package combination for which the objects DLLs are displayed.
Sequence	This is a display field. It links a sequence number to the conversion dll objects.
Object DLL's	The conversion scripts currently available in the vrc structure of the package combination in the system (located in the obcdll directories). A conversion dll has this naming convention: odll<t><pp><mmm><nnn> t = Type of the script f-script → Populate one target table m-script → Populate n target tables pp = package of the base target table mmm = module of the base target table nnn = table number of base target table **Example f-script: odllftcmcs045:** This is the migration mapping object for target table tcmcs045 **Example m-script: odllmtcmcs003:** This is the migration mapping object for target tables tcmcs003 and whwmd200. In this case tcmcs003 is the base table.
Commands	Description
Generate	Use this option to fill the available conversion objects within the vrc structure your package combination. Before generation fist the table is cleared and then generated

Fields	Description
	Note: You cannot manually maintain the information about available conversion script objects generated by the DCA Studio.
Delete all	Use this option to clear the table.
Cancel	Closes the session.

History/Multiple dumping Process per Target Table (bcgct8571m000)

This session and the related sub-sessions are used to split mapping objects to:

- Migrate history/static data before live migration. It reduces the system downtime during the live migration. First migrate the static data of a migration object up to a specific date and during the Live migration only run the migration object from that specific date up to current date. Splitting the migration object in 2 ranges.
- Migrate a source table which contains a large amount of data. To lower the total execution time of the mapping object it can be split into multiple objects where each object migrates a specified range of the main source table.

For more detailed information how to set up this functionality, see **Error! Reference source not found.** Split mapping objects using History/multiple dumping process per table

You cannot add any information in this session. The **history/multi-process per table** setup is initially performed in the DCA Studio. During the conversion build you must enable the **Consider History and Multi process setup** flag in the **Settings** option of your migration profile. When you perform a conversion build, a **splitinfo** file is created in your BaanIVc4 Engine environment, where the dll objects are located (By default *$BSE/application/<Your VRC>/obcdll*). Therefore, after starting this session, you must import information from this file by clicking ***Specific → Import from DCA generated file***.

This table shows the fields and commands in this session:

Fields	Description
Target Table	The target table to which you migrate the history data. This is a display field.
Number of Processes	This is a display field, and indicates the number of processes used to migrate data to the target table. To change the number of process, use the option Specific → Change No. of Processes For migrating the static history data use 1 process and for splitting a mapping object use a value greater than 1. **Note:** By default, during import this field gets the number of processes you have maintained in Conversion Parameters (bcgct0100m000) session

Fields	Description
Migrate	This is a display field. And provides the information about the source table from which data is being migrated. You can compare this with the migrate <source table> statement in your conversion objects maintained in DCA Studio.
Split data on fields	This is a display field. Shows the fields on which you are going to split or group the source data. **Note**: While splitting data, you can use the primary index of the source table, or some date fields. This setup is initially performed in the DCA Studio, you cannot modify it.
Selection Criteria tab	The tab(s) are enabled only if you are migrating history data and want to use multiple processes. Note: The scenario is not advisable, if your split criterion is not based on a unique index. You must specify the range, you like to consider. These values are used for generating the boundary values automatically for the processes.
Commands	Description
Import from DCA generated file	This is the first action you must perform, when you start this session. The multi-process or history data migration setup you carry out in the studio during a conversion build generates a splitinfo file. This option reads the splitinfo file and populates the data in this session.
Change Number of processes	Use this option, to use multiple processes during data dumping for a target table. When you increase the number of processes, ensure that you have adequate hardware resources to support the load.
Generate Range values per processes	This option calculates the range values for each of the processes, if you are using multiple processes to dump data. This action counts the number for records in the selection criteria specified in the Selection Range per Source Fields (bcgct8570m000) and calculates the boundary values for each process. Caution: This action takes long time if the source tables contain many records. Instead of generating you can also maintain the rage values in Range values per processes session.
Open Range values per Processes	This option starts the “Data Selection range per Group per Target Table” (bcdmp8572m000) session. Within this session you can maintain the ranges per source table.

Data Selection range per Group per Target Table (bcdmp8572m000)

This is the session you can define the “from” and “to” ranges for each data dumping process. It is required to maintain the range of values for each process. To automatically generate the values, select “Generate Selection Range Values” from the Specific menu. The automatic generation takes time to scan the entire source table data and split across the number of process. It is also possible to manually maintain data in this session.

Currently you cannot manually maintain the selection criteria for each process (you must use GTM in case you want to manually maintain the data). If the data is maintained manually and you use multiple processes to dump the data, the duplication of values can occur in the created data files. Different processes can dump the same source records in their respective data files. These duplicate values can stop loading of the data to Infor LN. To resolve this problem, you can generate the values during the test migration run, and use the values for manual entry when you actually carry out the migration process.

This screen displays the range values specified to extract data from source table *tfacp200 (Open Items),* when you use **two parallel processes**. During run-time, the range values are taken as follows:

For example if you must split the tfacp200 mapping object. Assume you set the number of processes to 2 and you used the full primary index.

In the range values per process you specify 3 values:

{‘’,’’, 0, 0, ’’, 0, 0}

{‘ 1001’, ‘ACP’, 19960358, 0, ‘’, 0, 0}\

{‘YASH02’, ‘ACP’, 19961449, 0, ‘’, 0, 0 }

This leads to these ranges for the 2 processes:

1st Process → Take values between {‘’,’’, 0, 0, ’’, 0, 0} and {‘ 1001’, ‘ACP’, 19960358, 0, ‘’, 0, 0}

2nd Process → Take values between > {‘ 1001’, ‘ACP’, 19960358, 0, ‘’, 0, 0} and {‘YASH02’, ‘ACP’, 19961449, 0, ‘’, 0, 0 }

This table shows the fields and commands in this session:

Fields	Description
Target Table	This is a display field, and indicates the target table to which you migrate the history data or use it to split the object over a number of processes to dump the data.
Source Table	The source table.
Split on Fields	The fields, on which the source table must be split. This is defined in the DCA studio and imported via the “splitinfo” file

Field values	The range values for each of the processes. For example, if you use two processes to dump the source table, you can see the range values as follows: 1 {} |* The from value for 1st process 2 {up to value for 1st process} 3 {up to value for the 2nd process} Note: You can generate these values using the option Special → Generate range values per processes.
Commands	**Description**
Generate Selection range values	Use this option to automatically generate the range values. The process takes time, because it must scan the entire source table data split across the number of process.

List of Target tables being converted (bcgct9188m000)

In this session you can maintain the “list of tables to be migrated” which is used by the dump engine. The first step is to generate the list based upon the mapping objects defined in the vrc structure of the package combination, see also session DCA Generated Conversion DLL objects (bcgct0106m000). The generation uses the “mscripts” file which is delivered as part of the logic pack or created during a conversion build from the DCA studio. This file is used to also show the related tables which are part of a m-script type mapping object.

The second step is to synchronize the “list of tables” with the “tables by batches” list from the LN Engine. You can synchronize automatically or manual.

All the object with the field “Skip” set to “No” are used in the dumping process. The field can only be set at the LN engine in the “tables by batches” which is synchronized with the current session. During the synchronization also tables can be imported into the list for which no conversion object is present. In that case the Skip field is automatically set to skip. After a manual import the data needs also to be exported and imported at the tables by batches in the LN Engine.

Within this session you can also specify a priority for a mapping object. By setting the priority you define the sequence how the mapping objects are assigned to the dump engine processes.

Note 1: You must maintain this information in all the source companies from which you want to migrate data.

Note 2: If you plan to use history data migration or number of parallel dumping processes for each target table (when source data size is large), you must perform the respective setup (History/Multiple dumping Process per Target Table (bcgct8571m000)), before you generate information in this table.

This table shows the fields and commands in this session:

Fields	Description
Company	The current company number for which you are generating tables to be migrated. This is a display field. **Note**: The data dumping process copies the data from other companies when you migrate multiple source companies to your current company from which you start the dumping process.
Target table	The target table name. The standard format used to represent target table is: ppmmmnnn Wherein: pp represents: Package mmm represents: Module nnn represents: Table number Example: tccom000, tttxt010
Skip?	This is a display field indicating if a Target table needs to be migrated or skipped during dumping of the data. The field is set during the synchronization process with the LN Engine. At the LN Engine you can specify which tables needs to be skipped or not. If the table is selected at the LN Engine but at the BaanIVc4 Engine there is no Mapping object then also the flag “Skip” is set to “Yes” by the BaanIVc4 Engine. **Note**: For m-Objects the “Skip” flag is also visible for all related tables in the related table session (bcgct9585s000).For more information see the Related tables.
Object Present?	Indicates whether a corresponding conversion script is present in the system for the current package combination. During the “batch list” synchronization process this “skip” flag is set to “Yes” when there is no conversion object in the system.
Priority	You can specify the priority to indicate the order of the dumping. If you do not indicate the priority, the dumping process first dumps the tables having Duration field value longer and then the tables with Duration field value Shorter. When you do not specify the priority, the tables with longer durations get the precedence over the shorter durations.
Duration	This field gives the information about the data size of a target table. Following are the possible values: • Shorter: The data file is small in size. • Longer: The data file is big in size.

Fields	Description
	You can set this value in LN Engine, and it is made available during synchronization process between the source and the target Engine. The dumping process first selects the tables with longer duration.
Related Tables	Indicates whether a target table is an M-Object (See Terms & Abbreviations). The target table is an M-Object, when the value of this field is greater than one, and the current target table is the base table. To view the related table, on the Special menu, click Related Tables.
Logical table?	Indicates whether the target table is a logical table in the current LN company for which you are performing the migration. For logical tables, the field Skip is set to Yes automatically. This information is synchronized from LN Engine.
Process Number	In case you use multiple processes to create data dumps for a target table, that target table occurs as many times as the number of process. For example, you have planned to use 4 processes to create data files for the table tfgld100, then you can find 4 records for table tfgld100, each one with process numbers 1, 2, 3, 4.
Commands	Description
Generate	Use this option to populate all the DLL objects placed by the DCA Studio module. Additionally, this option inserts the list of object DLLs in the bcgct988 table. It scans the directory obcdll in all the Package VRCs. The scanning starts from the current VRC of the package combination that pertains to the login and progresses to all the VRCs that are derived from the current Package VRC. For M-Scripts, a file with the name mscripts must be present in the obcdll directory. This file is delivered with the installed logic pack or can be created by performing a conversion build from the DCA Studio. Based on the mscript file the related child tables per base conversion object table are stored in the “Child Tables for repeated I/O” (bcgct985) table.
Related Tables	This option starts the “Related tables of a M-Script” (bcgct9585s000) session, which displays the child (related) tables for the M-script mapping object.
Import Target Details	This option can be used to synchronize the batch list from the Target side. The option is available for all scenarios. But in case of “Single machine” or “Two Machine with client server connection” the batch list can be synchronized also automatically from the LN Engine side. Before importing the Target details it needs to be exported from the LN Engine. The export at the LN Side creates a <compnr>.logical.988.info file in the defined “Directory for Engine Files”. In case this directory is not shared, between the source engine

Fields	Description
	and the target engine, this file needs to be copied to the source environment's "Directory for Engine Files". The file contains information about target tables to be migrated, the logical/physical mapping, bypass(skip), duration, mode of loading information which is imported into the source environment.
Export Source Details	This option can be used to synchronize the batch list from the Source side to the target side. The option is available for all scenarios. But in case of "Single machine" or "Two Machine with client server connection" the batch list can be synchronized also automatically from the LN Engine side. You can export the source information after you perform the Import tables from file process. This action creates an export file in the "Directory for Engine Files" specified in the parameter (bcgct000) table. The name of the file is "<company>.objects.info file". If the "Directory for Engine Files" is not shared, between the source engine and the target engine, you must move the file manually tot the "Directory for Engine Files" specified for the target side. You can import the file from the 'Data Load Dashboard" (bcgcteasym000) via the specific menu option at the LN Side.
Generate Priority from Simulation	You can calculate the priority of data dumping from the data conversion simulation (record counting) information. The target tables with more number of records to be migrated get the highest priority.

Related Tables of a M-Object (bcgct9585m000)

Use this session to view the related tables linked to the parent (base) migration object. Migration M-objects are related to a base target migration table but dumps out also data for other target tables. When the objects are retrieved via the generate option in the "List of Target Tables to be Migrated" (bcgct9188m000) session, the content of the mscript file is analyzed to get the related table information.

This table shows the fields of the session.

Fields	Description
Parent Table	This field displays the parent (base) conversion object table.
Company No.	This field displays the current company number.
Child Table	This field display which child table is related to the parent (base) conversion object table. During carrying out the Parent (base) conversion object also data is dumped out for the related child tables.

Fields	Description
Skip?	This is a display field indicating if a child table needs to be migrated or skipped during dumping of the data. For more details, see the “Skip?” field in the “List of Target Tables to be Migrated” (bcgct9188m000) session.
Commands	Description

Validate Data Integrity (bccnv4233m000)

This session uses the bdbvalidate6.2 utility of the Baan Tool. Find this utility in *$BSE/bin* on your Infor BaanIVc4 Server. This session presents a front-end to use this utility.

This is an optional session and allows you to do a data validation prior to the migration. At a minimum the data validation at the Target side after migration of the data is required.

For more details about the validation of the data integrity, see **Error! Reference source not found.**: **Error! Reference source not found.**

This table shows the fields and commands in the session:

Fields	Description
Package Combination	Enter the package combination that is linked to the companies for which you want to validate the data integrity. The default value for this field is your working package combination.
Company	Enter the range of companies to validate the data integrity. Ensure that, these companies are linked to the specified package combination.
Package	Enter the package combination for which you want to validate the data integrity. Ensure that, these companies are linked to the above package combination.
Module	Enter the module for which you want to validate the data integrity.
Table	Enter the table of the selected module for which you want to validate the data integrity.
Maximum No of errors	Enter the maximum number of errors that are allowed for each table. In case the errors in the table exceed the specified value, the scanning of that table is stopped.
Error Report	Select this check box to print the error report.
Path Report files	The report is generated in the specified directory.

Fields	Description
Commands	Description
Continue	Start the validation process for the range of values entered in this session.

Conversion Codes (bcgct9190m000)

Use this session to maintain Codes to recode your master data during migration. For more detailed steps on 'how to recode your data', refer **Error! Reference source not found.** : Recoding

Note 1: There are two conversion codes which are standard used and cannot be used for any other recoding purpose.

- "*COMP"* code is reserved for cross company migration.
- *"UNIC"* code is reserved for the Unicode correction feature.
- *"ITEM"* code is reserved for the Item recoding feature.

Note 2: To migrate several source companies, you must maintain the recoding information in all the companies, even if the master data is stored in one company. Alternatively, you can share the bcgct990 and bcgct991 tables in your current working company

This table shows the fields and commands in the session:

Fields	Description
Code	You can specify a code, to identify your recoding scenario. The codes "COMP", "UNIC" and "ITEM" are reserved of which conversion code "COMP' is created by default.
Description	Specify the description that explains your recoding scenario.
Values Present	Indicates whether the conversion code has at least one applicable value present in the "Values by Conversion Codes" (bcgct9191m000) session.
Commands	Description
Values by Conversion Code	On selecting this option, you can start the "Values by Conversion Codes" (bcgct9191m000), in this session you maintain the mapping between your existing value and new value in the LN environment.

Values by Conversion Codes (bcgct9191m000)

You can start this session from the Special menu of Maintain Conversion Codes (bcgct9190m000) session. In this session, you can maintain the mapping between the old codes in your Infor BaanIVc4 environment and the new conversion codes in your new Infor LN Environment.

The old value and new values can be created manually, retrieved via the source master data table or via a dll function, in case you have a specific algorithm to generate the new value from the old value.

To remove all the values by conversion codes you can delete the conversion code.

This table shows the fields and commands in the session:

Fields	Description
Code	This is the conversion code as specified in the conversion codes session for which you want to maintain the old and new values.
Old value	Enter the existing codes as present in your source environment. For example, if you are recoding your warehouses, you must enter the existing warehouse codes that you like to recode.
New value	You must mention the corresponding new values for the old codes. It is not required to specify the new values for all the old codes mentioned in the Old Value field; However, using empty New Values may result in data corruption problems after the data conversion.
Apply	Indicates whether the Old Value must to be replaced with the New Value. If the New Value is not empty, this ss read-only and always Yes. If the New Value is empty, this field is editable and can be changed to Yes, which means that the Old Value is replaced with an empty value or changed to No.
Commands	**Description**
Old Values from Table Field	Use this option to fill the codes automatically if you know the master data table where the codes are located. Entries are created with an empty New Value and Apply set to “No”. Replace the empty New Value with a desired New Value. This will automatically change Apply to “Yes”. To replace the Old Value with an empty value, leave the New Value empty and change Apply to “Yes”. This option starts the “Old Values from a Table Field” (bcgct9291m100) session.
New Values Using a Dll Function	If you have some specific algorithm to populate the new values, you can code that function in a DLL. Use this option to call the dll to populate the new values. For new values that are not empty, Apply is defaulted to “Yes”. For more information refer the session Fill new values using a dll function (bcgct9291m200).

Old Values from a table field (bcgct9291m100)

You can start this session from the Special menu of the Values by Conversion Codes (bcgct9191m000) session. Use this session to match the old codes in the existing source environment, to the corresponding conversion code. The values are retrieved from your current company.

This table shows the fields and commands in the session:

Fields	Description
Table Field Name	Enter the table field, from which you want to fill the old values. The coding standard for the table field is 'ppmmmnnn.ffff' pp → Package mmm → Module nnn → Table number ffff → field name Example: tcmcs003.cwar
Filter Criteria	Specify the filter criteria for the value before it is added to this table. For example, to recode only the warehouse codes starting with "AB", so you can specify the criteria as tcmcs003.cwar like "AB.*".
Commands	Description
Continue	Click this button to import the old values from the table field. Existing values are retained.

New Values using DLL function (bcgct9291m200)

This session automatically generates the new values for the corresponding old master data codes This is useful when you have some logic to generate the new value. In that case, you must create a dll function, in which you can program your logic. By default, BaanIVc4 Engine provides the bccnvdll9000 dll for this purpose. You can create your own dll, if you do not have source codes. The function in this dll can have any name, but the signature of the function is:

function extern domain bcstr30 bccnvdll9000.fill.cwar (

domain bcstr30 old-value)

The naming conventionis:

<your dll name>.fill.<your conversion code>

For example a function look likes:

function extern domain bcstr30 bccnvdll9000.fill.cwar(

domain bcstr30 old-value)

{ |* Write your logic here. Here in all the old values,

```
        |*in front the letter “A” is added
                return ("A"& old-value)
}
```

Caution: Ensure that you compile this dll, without any errors.

This table shows the fields and commands in the session:

Fields	Description
Dll Name	Enter the dll name in which you have written the function to construct the new codes. This naming convention is used for the dll: ppmmm<dll name> pp → Package mmm → Module <dll name> → any user defined name Example: bccnvdll9000
Function Name	The function in the above dll where the actual logic is coded. When you specify the function, you need not specify any arguments. At the runtime the old values get passed to the function, one by one. Note: Ensure that you compile your above dll and created the object.
Commands	Description
Continue	Click this button, to call the function that constructs and populates the new value for each old value present in the Values by Conversion Codes.

Maintain Pre-Migration session (bcgct0103m000)

Use this session to hold pre-conversion sessions as developed by Infor via the standard Migration Tooling and Standard migration Logic Pack. You can import the file to this session, and run each session.

Note: After you start a session from this list, the system sets the “Executed?” field to “Yes”. You can also use this session to hold your custom pre-conversion sessions if any.

This table shows the fields and commands in this session:

Fields	Description
Session Name	Enter the code for the pre-conversion (BC session or application) session you must run as a pre-conversion activity. These sessions require a manual trigger and are not carried out automatically. This session is used to organize the pre-conversion sessions or you can consider it as a container holding all your pre-conversion sessions.
Description	The description of the pre-conversion session.
Executing Sequence	Displays the order in which the sessions must be carried out.
Select Yes/No	Set the field to YES, in case you like to skip the execution of this session.
Execute Yes/No	This field indicates whether the session is carried out or not. When you start a session from this list, the field is automatically set to YES. It does not guarantee that the session is processed, because you can cancel the session.
Commands	**Description**
Continue [Ctrl + Y]	Starts a wizard that displays the sessions you must carry out.
Execute all… [Ctrl + F2]	Use this option to view the sessions you want to carry out.
Import from ASCII file [Ctrl + P]	If the pre-conversion sessions are shipped in an ASCII file, this option imports that file. Note: The file structure is pre-defined. The file structure is shown in the Import Start-up sessions from ASCII file (bcgct0203s000) session.
Delete List	Deletes all the sessions from this session.
Execute Selected	Starts a session that you have selected.
Reorder	Use this option to arrange the sessions in multiples of 5 according to the execution sequence.

Import Start-up sessions from a Flat File (bcgct0104s000)

You can start this session from the Special menu of the Maintain Startup Sessions (bcgct0103m000). Use this session to import the pre-conversion sessions if they are shipped with the rules pack. This session retrieves the ASCII files in a pre-defined format.

A list of all pre conversion session can be created by printing all session for the bc package to a file and modify the file according the specified format. Another option is to retrieve the pre-migration sessions as defined for the Logic pack from the Migration Process checklist delivered with the Logic Pack and save them in a file with the specified format.

This table shows the fields and commands in this session:

Fields	Description
File Name	Specify the file that stores the details of this session. You must specify the complete path of the file. This file must contain the Session Code and Session Description, separated by a separator.
Separator	Specify the character that separates the session code and session description in the ASCII file.
Commands	Description
Continue	Imports the session details from the ASCII file. While importing the data, session are appended to the existing sessions.

To debug the generated script (bccnvdebug)

This session is used to start the DCA Studio generated conversion scripts in debug mode. The session is located in the Conversion setup -> Utilities menu.

This table shows the fields and commands in this session:

Fields	Description
Program Script	Specify the program script to debug DCA script. To start a conversion object in debug mode use this legend: <t><mapping object name> <t> = script type; “f” = for normal script and “m” for M-Script The mapping object name consists of: pp = Package (Target) mmm = Module (Target) xxx = table (Target) Example: ftcmcs001 or mtcmcs003 Note: For debugging a m-script mapping object the base conversion mapping table needs to be entered.
Commands	Description
Continue	Use this option to start the debugging process.

Note: When the BAAN_TZ_DIR BW environment variable is not set, the bccnvdebug program cannot use the time zone files to migrate BaanIV Dates. In such a case a warning message is displayed and the time zone is considered to be GMT-0

Conversion Process

Under this menu item you find sessions that help you to start the data dumping /extraction process. This table shows the the sessions that you must carry out before conversion, and some utilities can be used during data dumping process:

Session	Session Code
Procedure	Submenu
Dump Data (Create Data Files)	bcrun9087m000
Archive Data Dumping Run	bccnv5201m000
Check-list	Submenu
Archive Data Dumping Run	bccnv5201m000
Check currently logged-in users (Unix only)	bccnv5406s000
Check Environment of present Company	bccnv6400s000
Check Scheduled Jobs	bccnv5404s000
Validate Data Integrity	bccnv4233m000
Reports	Submenu
Print Dumping (data file creation) Status	bccnv9442m000
Dumping Error Logs (Functional)	bccun9599m000
Data Dumping Runs (History)	bccnv5500m000
Utilities	Submenu
Reset Conversion Process	bccnv0200s000
Generate List of Log Files to Analyze	bccnv5301m000
Analyze Log Files	bccnv5300m000

Dump Baan IV data (bccrun9087m000)

Run this session to start the data dumping process that creates data files, which can be loaded to your Infor LN Application. Before running this session all the required pre-migration activities as described in the data conversion process checklist must be performed.

Before starting the actual dump process the session “Range of Tables for Dumping (Scenario-1)” (bcrun4220m000) is started. In this session you can specify which range of tables needs to be dumped. Based upon the entered Range the dump process looks into the “List of Target tables to be converted” (bcgct9188m000) session. All tables falling within the range and have the “Skip” flag to “No” are taken into the dump process.

After you specify the range per package, module and table click the “OK” button to proceed with the dumping process. Click Cancel to exit the dumping process.

After proceeding with the dumping process a control question is asked if you are really sure to continue with the process. Click “No” to cancel the process or “Yes” to proceed.

If the dumping process was already started, you get a question whether a “fresh Conversion Run” is required or if to continue with the “Previous Run”. The latter option can be used when a dumping process was aborted but you do not want to restart the complete conversion run.

If the dumping process was already started also the required dumping directory for the exported data files is already present. A question is asked to keep the current directory and the contents or delete it and create it again. If you keep the directory it overrides only the existing data files if the files (tables) are part of the current dumping process. The location of the data files is determined by the “path of the data files” specified in the “Parameters” (bcgct0100m000) session. The path is extended with the current company number. For example, when your data files path is *$BSE/tmp/migration*, and the source company is *123*, then it creates a directory, *$BSE/tmp/migration/123*.

After you made a choice the dumping process continues and the main dumping process screen is displayed.

The main reason of the session is to monitor the dumping process. You can see how many dump engines are running and which Mapping object is running in the engine. After an engine is ready processing one mapping object it gets a new mapping object assigned until all mapping objects in the defined dump range are processed. The displayed progress indicator does not represent the progress in terms of volume of data dumped but, indicates the progress in terms of the number of tables dumped.

This main dumping process screen displays this information:

Information	Description
Total tables to dump (Row)	The total number of tables in the selection range you entered. Considered the ones within the range which are set the be skipped.
Tables dumped (Row)	This field displays the number of tables for which the data files are created.
Dumping Tables (Column)	This column displays the Mapping object running in the engine. For the standard objects this is also the dump for the table. In case of an M-object the table specified is the main table. The related tables for the m-script mapping object are not shown. The number of rows corresponds to number of dumping processes. (up to 19 engines)
Company No. (Column)	This column displays the number of the company to which the data is being dumped. The number of rows corresponds to number of dumping processes. (up to 19 engines)
Records dumped (column)	This column displays the records dumped so far into each target table in the current dumping process and considering the refresh frequency

Information	Description
	for number of records as specified in the “Parameters” (bcgct0100m000). The number of rows corresponds to number of dumping processes. (up to 19 engines).

The table shows the possible actions you can perform from this screen. You must use the respective key combinations to carry out these actions.

Actions	Descriptions
Cancel dumping [Ctrl + A]	This action stops the dumping. When you use this key combination, the system prompts you for the confirmation. Click Yes to stop the process. The system waits for the current dumping processes to complete.
Invoke bshell [Ctrl + P]	Use this key combination to start additional dumping processes.
Revoke bshell [Ctrl + N]	Use this key combination to reduce the number of processes in case of system overload.

Attention points

Reset the conversion process

When you start the dump process session, a message may appear “Dumping started already by <user> in company <company number>”. The message is displayed when somebody else has started the conversion run or the conversion run you have started earlier got killed. Ensure that no user is performing a conversion and the last conversion is terminated. You can reset the lock to start the conversion run again by using the session, *Conversion Process → Utilities → Reset Conversion Process (bccnv0200s000)*

After you reset the conversion process you must restart the dumping process and a message is displayed to start a new conversion or continue processing of the not dumped tables (completed mapping objects) of the previous run. When you choose the “Fresh Run” option you can specify the new range of tables to dump.

After resetting the conversion you must archive the dumping history manually if want to keep the dumping history.

Number of dump engines

The number of dumping processes that you can invoke is the number of user licenses you have minus 1. The maximum number of dumping engines is 19. Always evaluate the system resources to

set the amount of dumping engines. Too much dumping engines can cause to a lower performance and increases the total execution time.

Data Files

During the dumping process, data files (.dat) are generated for each of the Infor LN tables within the range you have specified. If the target data base is MSSQL and the dump engine is ready with processing a mapping object all the dumped data files are transformed to *.da2 files.

Data files in ASCII format are created in the path specified for Directory of data files field in the Parameters (bcgct0100m000) session. For example:

```
<Directory of data files>/<company number>
<package+module>/<tablename.dat>...
```

```
The data file for the tcccp000 table is available in the following location:
```

```
< Directory of data files>/<company number for which conversion is being
run>/tcccp/tcccp000.dat
```

Archive Data Dumping Run (bccnv5201m000)

Use this session to archive the details of the last dumping process. In case you do not archive the details, the next conversion run overwrites the present information. Hence you must archive the dumping information, after data dumping or resetting the conversion process.

In Conversion Parameter (bcgct0100m000) session, the ‘Archive time logs after conversion?’ field must be set to “Yes”, to archive the time logs automatically, if the conversion run ends smoothly. If the conversion process got killed then archiving of the conversion logs must be handled manually.

In the session you can specify any reference information about the last conversion run. You can use this information to interpret the archived information. Click **Continue** to start the archiving of the dump information.

This table shows the fields and commands in this session:

Fields	Description
Enter the reference Text, if any	Use this field to add a reference text to for the archive dumping time log record.
Commands	**Description**
Archive	This option stores the Dumping time log data.

Check Currently logged in users (Unix only) (bccnv5406s000)

Use this session to see which users are currently connected to the system and to which specific package combination. The list is presented in a report.

You can use this session to see which users are connected to the system at the moment.

Check Environment of present Company (bccnv6400m000)

Before you perform the conversion run, you must check the hardware and software configurations at the source. Use this session to retrieve the information from the system. The information is shown in a report.

The session provides this information:

- User environment variables on the source system.
- Source Porting set information.
- Data size of the source company.
- Free space available for data dumps.
- Information about the creation of all the BC tables.

This table shows the fields and commands in this session:

Fields	Description
Baan IV Porting Set	Set the field to Yes to view the porting set details of the current Infor BaanIVc4 Environment.
User Details	Select the option to view the report of the conversion user details.
Target Server name	This field is available only when your conversion scenario is Two machines with client server and the operating system of Infor BaanIVc4 environment is UNIX. This information is used to get the FAT client details.
Path of Baan BaanIVc4 client	This field is available only when your conversion scenario is Two machines with client server and the operating system of Infor BaanIVc4 environment is UNIX. This information is used to get the FAT client
Check BC tables	Set the field to Yes if you want the system to check whether the BC package tables are created properly.
Source company size	Set the field to Yes to calculate and report the current company size to the user. This process can take some time, depending on the volume of the data.

Fields	Description
Free space on dumping directory	Set the field to Yes, to report the free space on your data files directory.
Commands	Description
Continue	Select this option to start analyzing.

Note: If this session is not displayed properly, ensure that, you have entered "*bcyesno*" as domain in the Baan Tool's session Domains to be Displayed as Checkbox (ttgfd4520m000).

Check Scheduled jobs (bccnv5404s000)

The session generates a report of scheduled jobs along with the next execution date and time. Use this session to check if there are jobs scheduled to run during a scheduled dumping process. Jobs can slow down the the dumping process. The results are displayed as a report.

Analyze Scheduled Jobs is developed on these UNIX versions:

- HP UNIX : Version HPUX10.20 & Version HPUX11.00
- Sun Solaris: Version 2.5.1

Note: The smooth functioning of this utility on other flavors of UNIX is not guaranteed due to difference in the methodology of each UNIX flavor. This utility works well on Windows NT. For more details see these manuals:

- UNIX Manual for Crontab and AT commands
- NT Manual for Task-Scheduler and AT commands
- Job Manual for Baan

This table shows the fields and commands in this session:

Fields	Description
Time (Start< > End)	Enter the time range for which you want to check the execution time of Job. During data conversion, ensure that no other scheduled jobs are run, because that reduces the system performance. You must enter the time in Hours format.
Report Operating System Jobs?	Set the field to Yes to analyze the operating system - level job scheduled in the specified time frame.
Report Baan Jobs?	Set the field to Yes to report all the jobs scheduled in InforBaanIVc4.
Report Details of Jobs in all companies?	Set the field to Yes to scan and report jobs scheduled in all the companies of Infor BaanIVc4 application.
Commands	Description
Continue	Use this option to select a device for printing.

Print Dumping Status (bccnv9442m00)

The purpose of this session is to print the table details (progress), during the data dumping process. Use this session to find out the target tables' data files that are created, and are in progress.

The result is displayed as a report.

This table shows the fields and commands in this session:

Fields	Description
Dumped	This field is used to specify if the report must contain either the tables that are dumped or the tables that are not dumped.
Table name (From<> To)	Specify the range of tables for which the dumping process is performed.
Commands	Description
Continue	Use this option to select a device for printing.

Functional error Logs during dumping (bcrun9599m000)

Use this session to view the functional errors logged during execution of the mapping objects in the dump process. For instance, if a mapping objects requires data from a pre-migration mapping table but the record is not found a message is logged in the "Error logs" (bcrun999) table.

Error messages during the dumping process are only logged if in the Maintain conversion Parameters (bcstc0100m000) session the "Error Logging" is set to "Yes". One exception to this rule is when in the migration mapping object it is stated to always log a message regardless of this setting. In a test migration it is advised to log the error messages and analyse them. During the Go-Live migration run it is not advised to use the error logging as it consumes performance.

The messages in this session are only displayed when in the parameters the "Archive error logging" in the parameter (bccnv0100m000) session is set to "No". If this parameter is set to "Yes" the error log messages are archived and must be viewed using the "Error Logs – archive" (bccnv5503m000) session.

This table shows the fields and commands in this session:

Fields	Description
Company Number	Displays the company number for which the error has occurred.
Table Name	The target table for which the error occurred.
Group Number	The process in which the errors have occurred.
Sequence	The sequence number of the logged error message.
Error Description	The logged error message for the dumped table.

Fields	Description
Commands	Description
Clear all logs…	Use this option to clear all the logs present in this session.

Data dumping Runs History (bccnv5500m000)

Use this session to see the run codes with the start and end times of the dumping process per run. After selecting a specific run number you can use the specific menu to drill down to the Conversion time Logs per run and the archived error message for the selected run.

To clear the Data dumping runs you can select the run and delete it. All related data is deleted as well.

This table shows the fields and commands in this session:

Fields	Description
Run Code	Each time a new dump process is started also a new run code is created.
Start time	The start time of the dumping process.
End time	The end time of the dumping process.
Elapsed time	The total elapsed time of the dumping process.
Comments	Displaying when the run was archived.
Commands	Description
Open Dumping Logs	Use this option from the specific menu to start the Conversion time log for the selected run. This option starts the “Display History – Conversion Time log” (bccnv5501m000) session.
Open Error logs	Use this option from the specific menu to view the archived error log messages for the selected run. This option starts the “Error Logs - Archive” (bccnv5503m000) session.
Print conversion Time Log	Use this option from the file -> print menu to print the detailed information of a selected conversion run or a range of runs.

Display History – Conversion Time log (bccnv5501m000)

Use this session to view the details of a specific conversion run.

This table shows the fields and commands in this session:

Fields	Description
Run Code	The Run code for which you want to see the conversion time log details.
Company	The company for which the run is processed.
Table	The name of the dumped target table.
Elapsed Time	The time required to dump the data for the target table.
Records Dumped	The number of records dumped for the table.
Process Number	The Process number in which the data for the table was running. Used when the migration mapping object is split over multiple dumping (process) engines.

Error Logs - Archive (bccnv5503m000)

Use this session to view the archived error log message for a dumping run.

This table shows the fields and commands in this session:

Fields	Description
Run Code	The Run code for which you want to see the conversion time log details.
Company	The company for which the run is processed.
Script	The name of the script for which the error message is generated.
Error Description	The archived functional error message logged for the processed script within the dumping process run.
Process Number	The Process number in which the data for the table was running. Used when the migration mapping object is split over multiple dumping (process) engines.

Print History – Conversion Time Log (bccnv5401m000)

Use this session to print all the conversion details of a specified range. The conversion time log can be printed in the order of the tables or in the order off processing time for the table. The results are displayed in a report.

The conversion run can be used to compare conversion runs or can be used to set the priorities for migration mapping objects with long processing times.

Note: It is recommended for each complete conversion run to print the conversion time (to file).

This table shows the fields and commands in this session:

Fields	Description
Run Code	Specify the Run code from and to range for which the conversion time log details must be printed.
Table	Specify the Table from and to range for which the conversion time log details must be printed.
Company	Specify the Company from and to range for which the conversion time log details must be printed.
Sort mode	The conversion time log can be printed in two sort modes: 1 According the table names in alphabetical order 2 In an elapsed time order with processing time of zero is at the top.

Reset Conversion Parameters (bccnv0200s000)

The data dumping process locks the conversion parameters so that no other conversion run can be handled for a company at the same time. However, there are situations when, you encounter some problems such as non-availability of disk space, performance/memory overload, restart of the environment/database, the conversion script in debug mode and so on. These problems make the data dumping process non-responsive. The only solution to this situation is to kill the dumping process. However, the locks on the conversion parameters are not removed.

Use this session to release the lock on the conversion parameter. If the lock is present, the Continue button is enabled. Click Continue to release the lock.

Generate List of Log files to analyze (bccnv5301m000)

Use this session to create files used by the “Analyze log files” (bccnv5300m000) session. This utility is available only if InforBaanIVc4’s operating system is *UNIX*.

Click “Yes” to generate fresh log files.

Analyse log files (bccnv5300m000)

Use this session to analyze the log file generated by the source engine, compile the information in a file and mail it to the user’s mailbox, specified in the parameters (bccnv000). This utility is available only if the BaanIVc4 Engine is installed on a ***UNIX*** machine and for a two server scenario without a client connection.

This session takes some processor time, to perform the periodical checks. You must not use this utility during the Go-Live migration run.

This table shows the fields and commands in this session:

Fields	Description
Analyze Log Files Active (y/n)?	The field displays whether this utility is running or not. The field contains one of these values: Yes: The Analyze Log File utility is activated. No: The Analyze Log File utility is not activated.
Commands	**Description**
Start Analysis	Use this option to start the utility. A background process that starts analyzing the log files is initiated.
Stop Analysis	Use this option to stop this utility. The background process stops.

Note: To create the files used by this session, you must first run the “Generate list of files to analyse” (bccnv5301m000) session.

Conversion Simulation (Optional)

Under this menu item you find sessions to run a simulation of the migration mapping objects. The simulation process uses special migration object version to start a record count. The record count and the estimate process time can be used to set the conversion priorities.

Prior to running these session a migration build from the DCA needs to be generated. Within the Generation settings of the DCA studio the objects needs to generate in simulation mode. This adds the record counting code to the migration mapping objects.

Special generated objects are stored in the vrc path in the “/obcsml” directory. The new generated objects do not impact the installed “/obcdll” objects during the installation of the logic pack.

This table shows the available sessions in this section:

Session	Session Code
Source Companies to be Migrated (*)	bcgct9189m000
Simulate Data Conversion	bccnv8101m000
Simulation Run Details	bccnv8502m000
Display History of Simulation Run	bccnv8503m000

(*) This session is described in detail earlier in this chapter.

Simulate Data Conversion (bccnv8101m000)

Use this session to retrieve all the generated simulation objects by the DCA Studio. Per simulation object you can set the “Skip” flag as to mark to object to be used in a simulation process or not. Within the session you can also set a range of objects to “Skip” or not. After you specified the simulation objects you can start the simulation.

This table shows the fields and commands in this session:

Fields	Description
Company number	The company for which a simulation run is required.
Target Table	The target table for which you want to a simulation.
Skip Yes/No	Does this target table needs to be considered in the simulation run or not.
Commands	**Description**
Generate	This option retrieves all generated simulation objects by the DCA Studio from the “/obcsml” directory.
Skip Yes/No	The option starts a sub session “Set Skip Yes/No to Range of tables” (bccnv8201s000). Within the session you can set the “Skip” flag for a range of target tables. You can enter a selection range based upon package, module and table.
Execute Simulation	This options starts the simulation run. It takes only the target tables marked with the “Skip” flag set to “Yes” into account.

Simulation Run Details (bccnv8502m000)

In this session you can view the simulation results generated by the previous described session. You can view the record counting for a source table in a specific company. You can view the start date and end date and the processing time for the simulation.

The current simulation results can be stored in a history table to compare different simulation results. If the data is not stored in history the data is deleted and recreated for the new simulation.

This table shows the fields and commands in this session:

Fields	Description
Company number	The company for which a simulation run is done.
Target Table	The target table a simulation run is done.
Source table	The used source table.
Records in Source	The number of records in the source table.

Number of records that migrate to target	The number of records that are migrated to the target table.

Commands	Description
Move to History	This option stores the current simulation results in a History table.

Display History of Simulation Run (bccnv8503m000)

Use this session to view the simulation results of an earlier simulation run. After the data is moved to history you have the option to print the data.

This table shows the fields and commands in this session:

Fields	Description
Simulation Run number	The simulation run number of the processed and move to history simulation run.
Company number	The company for which a simulation run is done.
Target Table	The target table a simulation run is done.
Start Date	The start date of the simulation run.
End Date	The end date of the simulation run.
Total Records to be Migrated	The total number of records to be migrated.
Source table	The used source table.
Records in Source	The number of records in the source table.
Number of records that migrate to target	The number of records that are migrated to the target table.

Commands	Description
Print History	This option starts a print session to print the simulation run details. The “Print History of Simulation” (bccnv8403m000) session is started.

Print History of Simulation (bccnv8403m000)

Use this session to print the stored history data of a simulation run. You can print the data based upon a range for company, Simulation Run and target table. The data is printed on a report.

This table shows the fields and commands in this session:

Fields	Description
Company number	You can specify the “from” and “to” range for the company number to print the data.
Run	You can specify the “from” and “to” range for the simulation run to print the data.
Target table	You can specify the “from” and “to” range for the target table to print the data.

Chapter 5 Special Migration Features

This chapter describes some special migration features and guides you through the special requirements that can occur during a data conversion project.

To copy data across companies

The next cases are considered to explain this scenario.

Case1: Cross company migration

For example, you want to migrate BaanIVc4 company ***500*** to LN company ***600***. To achieve this, you must follow the steps mentioned in Chapter 3. Additionally, you must maintain the company number mapping, before you perform the step ***4c.5***. To maintain the company number mapping, you must start the **Maintain Conversion Codes** (bcgct9190m00) session. After you start this session, you can view the conversion code **COMP**, which is reserved for mapping the source company to target company. You must select this record and click the **Open values by Conversion Codes** option, in the **Specific** menu. In this session, you must enter the old value as **500** and new value as **600**.

While maintaining the Conversion Parameters (bcgct0100m000), ensure that you have set the ***Copy to different target company?*** field to **YES.**

Note: Some of the application tables also contain the company number in their data. If this data is migrated to a different company in Infor LN environment, the data may not be processed properly. Therefore, you must correct the company numbers in the data during dumping. You can use the **Re-coding utility** in the DCA Studio to correct the data. The utility updates the conversion rules to handle the company number mappings you maintain later. The recoding utility is specific for a customer.

Case 2: Cross Company migration with multiple companies simultaneously

For example, you want to migrate BaanIVc4 company ***500*** to LN company ***600,*** and BaanIVc4 company ***200*** to LN company ***400*** together. To achieve this, you must understand the process for both the BaanIVc4 Engine and LN Engine explained in this table. See Chapter 4 for BaanIVc4 Engine details and Chapter 5 for LN Engine details. The activities with the same number, for example, 4***c.1*** and ***Ln.1***, can be performed simultaneously. It is assumed that you have already installed the BaanIVc4 and LN Engines in the respective environments. So in this case, the multiple companies are migrating like:

- 500 → 600 < From 500 to 600>

- 200 → 400 <From 200 to 400>

In this case, consider one pair as default companies, where the conversion run is started. In this example, you can consider 200 as the default BaanIVc4 company and 400 as the default Infor LN company.

BaanIVc4 Engine Process Steps	LN Engine Process Steps
4c.1. In the General User Data (ttaad2500m000), change your company to 500. And create runtime for the user data. Close the existing Baan Windows connection and reopen.	Ln.1. Change your company to 600. It is not required to change your user data.
4c.2. Maintain Parameters (bcgct0100m000). While maintaining parameters, ensure that you have set Copy to different target company? field to YES.	Ln.2. Maintain Parameters (bcgct0100m000). You can start this session from the Dash board (bcgcteasym000).
4c.3. Get DCA Generated Conversion Object dlls (bcgct0106m000). Note: You must use the Generate option in the Specific menu.	
4c.4. Generate Target tables that can be migrated from the above list (bcgct9188m000) Note: This option can be accessed from the Specific menu of the bcgct9188m000 session.	Ln.4. Generate OnpLoad Database (bcrun0207m000). This action is required when your LN Application's database is Informix. This session can be started from the Specific menu of the Batches (bcgct8188m000) session.
4c.5.You must map company 500 to 600 and 200 to 400 for the conversion code COMP In the Maintain Codes for Recoding data (bcgct9190m000) and Values by Conversion Codes (bcgct9191m000) sessions.. For more information on these sessions, see Chapter 4.	Ln.5. Maintain Batches (bcgct8188m000). You can start this session from the Dash board (bcgcteasym000) .
4c.6. Check Data Integrity (bccnv4233m000). The process checks for the unfilled required fields and the domain constraints violations. Note: This action is optional. However, verifying source data prior to the test conversion is the best practice.	Ln.6. Maintain Tables by Batch (bcgct8189m000). This session can be started from the Specific menu of the Batches (bcgct8188m000) session. .
4c.7. Carry out the applicable Pre-conversion sessions, in company 500.	Ln.7. Carry out the applicable Pre-conversion sessions, in company 600.
You must complete these activities in case you are dumping and loading simultaneously. Otherwise, skip this block and complete the steps in the Dump and Load later section	
	Ln.8. Synchronize with Source. In this process you synchronize the information maintained in

BaanIVc4 Engine Process Steps	LN Engine Process Steps
	the LN environment and Infor BaanIVc4 environment. You can access this option from the Specific menu of the Dash board. Pre-conversion (With Client Server) → Synchronize with Source.
4c.9. In the General User Data (ttaad2500m000), change your company to 200. And create runtime for the user data. Close the existing Baan Windows connection and reopen.	Ln.9. Change your company to 400. It is not required to change your user data.
4c.10. Maintain Parameters (bcgct0100m000). While maintaining parameters ensure that, you have set value YES for the Copy to different target company? field.	Ln.10. Maintain Parameters (bcgct0100m000). You can also maintain the parameters from the dashboard session.
4c.11. Maintain Source Companies to be migrated (bcgct9189m000). In this session, you must maintain companies 200 and 500. Note: This data must not be maintained in company 500.	
4c.12. Get DCA Generated Conversion Object dlls(bcgct0106m000) Note: You must select the Generate option from the Specific menu.	Ln.12. Generate OnpLoad Database (bcrun0207m000). This action is required only when your LN Application's database is Informix. You can start the session from the Specific menu of the Batches (bcgct8188m000) session.
4c.13. Generate target tables that can be migrated from the above list (bcgct9188m000). Note: You can access this option from the Specific menu of the bcgct9188m000 session.	Ln.13. Maintain Batches (bcgct8188m000). You can start this session from Dash board.
4c.14. You must map company 500 to 600 and 200 to 400 for the conversion code COMP In the Maintain Codes for Recoding data (bcgct9190m000) and Values by Conversion Codes (bcgct9191m000) sessions. For more information on these sessions, see Chapter 4.	Ln.14. Maintain the Tables by Batch (bcgct8189m000). Note: You can start this session from the Specific menu of the Maintain Batches (bcgct8188m000) session.
4c.15. Carry out the applicable Pre-conversion sessions, in company 200.	Ln.15. Carry out the applicable Pre-conversion sessions, in company 400.
4c.16. Check Data Integrity (bccnv4233m000). The process checks for the unfilled required fields and the domain constraints violations.	

BaanIVc4 Engine Process Steps	LN Engine Process Steps
Note: This action is optional. However, verifying source data prior to the test conversion is the best practice.	
	Ln.17. Synchronize with Source. In this action you synchronize the information maintained in LN Environment with Infor BaanIVc4 environment. Note: You can access this option from the Specific menu of the Maintain Batches (bcgct8188m000) session. Pre-conversion (With Client Server) → Synchronize with Source. Also ensure that all activities including 4c.16 & Ln.14 are carried out.
	Ln.18. Start the Conversion process. Note: You can start the session from the Specific menu of the Dash board. Conversion → Dump and Load.
4c.19. Start Dumping (bcrun9087m000). Note: The action in Ln.19 starts the menu browser of Infor BaanIVc4 Engine. From this menu browser you must start the dumping process, Data Conversion Adapter – Infor BaanIVc4 Engine → Conversion Process → Procedure → Dump Data (Create Data Files). When you start dumping for the company you can accept the full range. So that company 200 and 400 is considered.	
	Ln.20. Carry out the applicable Post Conversion Sessions, in company 400 & 600.
Dump and Load later: You must carry out these activities if: You extract the data (dumping) in the Infor BaanIVc4 environment and then load the data files to the Infor LN environment. In this case you can migrate one pair of companies at a time. Migrate company 500 → 600 first and then you can repeat the next process for companies 200 → 400.	
	Ln.8. Export target tables information. This action creates the <Target Company Number>.tables.info file in <directory of Engine files>/<Target Company Number> directory. The Directory of Engines files is maintained in LN Engine's Conversion Parameters. For example, if you have specified Directory of Engine files as /migration/share and your current target company is 600, then the tables are

BaanIVc4 Engine Process Steps	LN Engine Process Steps
	exported to the /migration/share//600/600.tables.info file. Note: You can access the option from the Specific menu of Dash Board (bcgcteasym000). Pre-conversion (Without Client Server) → Export Target Info. to Source. Ensure that you have carried out the Check Data Integrity and Maintain Batches session.
4c.9. Import Target Info. from file. This file (<target company number>.tables.info) is generated by action Ln.7. You must copy this file from the LN environment to <Directory of Engine files>/<your source company>location on the Infor BaanIVc4 environment. The Directory of Engine files is maintained in your BaanIVc4 Engine's parameters. For example, in case you specify Directory of Engine files as /migration/share and the source company to be migrated is 600, then copy the file as /migration/share/600/600.tables.info. Note: You can access this option from the Specific menu of List of Target tables to be migrated (bcgct9188m000) session. Ensure that all the activities including Ln.8 are performed.	
4c.10. Export Source Info.to file. This action creates the <Source Company Number>.objects.info, file in your <Directory for Engine files>/<Target Company Number>location. For example, if you have specified the Directory for Engine files as /migration/share and your current source company is 600, then the tables are exported to the /migration/share/600/600.object.info file Note: You can access this option from the Specific menu of the List of Target tables to be migrated (bcgct9188m000) session. Ensure that all the activities including 4c.9 are carried out.	
	Ln.11. Import Objects present information. The <target.company.number>.objects.info file is generated by action 4c.9. You must copy this file from the Infor BaanIVc4 environment to

BaanIVc4 Engine Process Steps	LN Engine Process Steps
	<Directory of Engine files>/<your target company> location on your Infor LN environment For example, in case you have specified your Directory for Engine files as /migration/share and your target company is 600, then copy the file as /migration/share/600/600.objects.info. Note: You can access this option from the Specific menu of Dash Board (bcgcteasym000). Pre-conversion (Without Client Server) → Import Source Info. to target.
4c.12. Dump Data (Create Data Files) (bcrun9087m000).	
	Ln.13. Transfer the data files generated as a result of the action 4c.12 to your Infor LN Server, in any directory and perform Load without dumping. You can access this option from the Specific menu of Dash Board (bcgcteasym000). Conversion → Only load. Note: In BaanIVc4 Engine, the created data files are organized in directories <package><module>. During the file transfer retain this directory structure. The Loader searches for the files in <Directory of data files>/600 path while loading.
	Ln.14. Carry out the applicable Post conversion session for company 600.

To migrate Text Tables (tttxt001, ttxt002, and tttxt010)

You must pay attention to the following points during the migration of the text tables.

- You must not maintain any conversion rules for the ***tttxt010*** table in the DCA Studio. By default, the object that dumps the data in this table to the ASCII file is shipped with the BaanIVc4 Engine.
- Ensure that you maintain the rules for ***tttxt001*** and ***tttxt002*** tables in the profile of DCA Studio, for which you perform the conversion build.

- When you define a batch in LN Engine, ensure that the batch contains ***tttxt001***, ***tttxt002***, and ***tttxt010*** text tables. During loading, the loader processes the data files in this specified order. The loader for table tttxt010 waits until table tttxt001 and table tttxt002 are loaded. If these tables are not specified the loader for tttxt010 waits indefinitely.

Split mapping objects using History/multiple dumping process per table

Use this feature if the table contains a lot of (static) history data and processing this table would take too much processing time during a final Go-Live Migration. Using the “History/multiple dumping process per table” feature you can reduce the system downtime during the live migration. First migrate the static data of a migration object up to a specific date at a convenient moment and during the final Go-Live migration only run the migration object from that specific date up to current date. This is splitting the migration mapping object in 2 objects/ranges.

Or use this feature if the source table contains too much data and you want to split the mapping object over multiple dumping processes. To lower the total execution time of the mapping object it can be split into multiple objects where each object migrates a specified range of the main source table and can be carried out in another dumping process.

For example, if the migration of tiitm001 table would take four hours to run in one dump engine process. If the total available engines were four than one engine would process for four hours and three engines would wait (not utilized). By splitting the mapping object in four objects/ranges each object can run in another dump engine process. So now each dump engine is utilized and handles a specific range of the source table. Now the lead time is one hour the total dump time for table tiitm001 remains the same four times one hour is four hours.

If you can only use 1 dump engine than splitting a mapping object in multiple processes does not lower the total migration dumping time.

In both cases, you require the DCA Studio to modify the standard delivered mapping object for a specified table.. You must complete the procedure steps before starting the actual conversion run.

For processing History data

1. Identify the field(s) in the source table which can help you recognize the history data. This can be a date field or a status field.
2. In Studio, go to Master Data → History & Multi-process.
3. Using the new button add the conversion object required for the process. Define the main query table and select the date or status field required (For more information refers to the *Data Conversion Adapter Studio 3.4 - User's Guide* (U9093F US).
4. In Studio, go to Conversion Rules Modeler → Conversion Build
5. In the Conversion Build screen use the Settings button. Onto the “XML and Script Info” tab select the “Consider History and MultiProcess Setup” option. In the build details specify the name of the mapping object (defined in step 2).

6 Run the conversion build. The conversion build creates a specific migration mapping object and a splitinfo file is created in your BaanIVc4 Engine environment, where the dll objects are located (By default $BSE/application/<Your VRC>/obcdll).

Continue in the BaanIVc4 Engine

7 Start the “History/Multiple dumping process per Target table” (bcgct8571m000). From the Specific menu select the “Import from DCA generated file**…”** option. This option reads the information from the splitinfo file created by the DCA Studio.

8 Select the “Change number of process” option from the Specific menu and enter the value 1 (it is safe to use one process for dumping). By doing this you specified that this specific mapping object is started in 1 dumping engine.

9 Select the “Open Range values per processes” option from the Specific menu. On starting the session, you find two processes. Against the first process enter the “from” value from which you like to select the data for migration. Against the second process enter the “to” value up to which you are migrating. The values must be enclosed in {}. If you are using a date value to split the data, you must enter the corresponding long number of the date

10 During the dumping process this mapping object exports (dumps) the records falling into the specified range.

11 Alter the range for the final Go-Live migration to migrate the remainder of the data.

12 Load the data for this table in append mode. Another option is to first create a sequential dump of the LN table then load the table (truncate mode). Then reload the data from the sequential dump.

For splitting the mapping object in multiple ranges

1 Identify the field(s) in the source table which can help you specify ranges for the mapping object.

2 In Studio, go to Master Data → History & Multi-process.

3 Using the new button add the conversion object required for the process. Define the main query table and select fields required. Most of the time it is a part of the main index of the source table. For more information refers to the *Data Conversion Adapter Studio 3.4 - User's Guide* (U9093F US.

4 In Studio, go to Conversion Rules Modeler → Conversion Build

5 In the Conversion Build screen use the Settings button. Onto the “XML and Script Info” tab select the “Consider History and MultiProcess Setup” option. In the build details specify the name of the mapping object (defined in step 2).

Note: By selecting the Multiple process option you cannot also select the 2GB option to set a maximum of data file size (unix environment).

6 Run the conversion build. The conversion build creates a specific migration mapping object and a splitinfo file is created in your BaanIVc4 Engine environment, where the dll objects are located (By default $BSE/application/<Your VRC>/obcdll).

Continue in the BaanIVc4 Engine

7 Start the “History/Multiple dumping process per Target table” (bcgct8571m000). From the Specific menu select the “Import from DCA generated file**…”** option. This option reads the information from the splitinfo file created by the DCA Studio.

8 Select the “Change number of process” option from the Specific menu and enter a value > 1. You can even choose a value higher than the maximum defined dump engines in the parameters (bcgct000) table. By this you just define the amount of ranges and each range is presented to an available dump engine process.

Note: If you are dumping on a Unix BaanIV environment and you are bound by the 2GB restriction than each specified range must result in a data file which does not exceeds this limit. If this is not the case extend the amount of processes or manually alter the ranges.

9 Select the “Open Range values per processes” option from the Specific menu. Within the session choose the “Generate Range Values per Process” option. This creates equal ranges based upon the amount of processes and the selection fields. If needed you can manually alter the specified ranges.

10 During the dumping process the mapping object exports (dumps) the records falling into the specified range and another defined range can be presented to another available engine.

11 During the load process the LN Engine evaluates the amount of available data files and loads the data in the de LN table.

Recoding

This feature enables you to recode master data, as a part of the data migration to Infor LN. During data migration, customers can choose to standardize their master data codes, for example, address codes, warehouse codes, and so on. During the migration the old source code is replaced by the new defined code.

Within the DCA Studio you must set up a conversion code, See also *Data Conversion Adapter 3.4 – Data Conversion Adapter Studio – User Guide (U9093F US)*

Using the DCA Recode Utility (part of the DCA studio installation) you can link domains to a conversion code. The recode utility searches through the specified logic pack and detects were the specified domain is used. On all places were the domain is used the mapping rule is modified and uses the predefined GetCode() function. During the dump process this function retrieves the new defined code based upon the old source code as specified in the Maintain Values by Conversion Codes (bcgct9191m000) session.

As the recoding functionality can apply to all mapping objects of a Rules Pack, the new value is not only replaced in the master data table but also on all the places where the master data field is present. For instance, the recoding of a warehouse code is not only done in the generic warehouse table but on all places where a warehouse code is used like sales orders, purchase orders, warehouse orders, Production orders, invoicing, and so on.

This list shows a short overview of the steps to use the recoding functionality.

- Define a Conversion code in the DCA Studio

- Create a specific Recode profile in the DCA Studio
- Use the Recode Utility of the DCA Studio to apply the recoding code to the complete logic rules and create the changed mapping objects in the new created recode profile.
- Define the recode code in Conversion Codes (bcgct9190m000) session in the source engine.
- Define the old / new values in the Maintain Values by Conversion Codes (bcgct9191m000) in the source engine.
- Dump and load the data.

The conversion codes can be defined in the Conversion Codes (bcgct9190m000) session. The defined conversion codes must be the same as defined in the DCA Studio.

These two predefined conversion codes cannot be altered:

- "*COMP"* code is reserved for cross company migration. For example, to migrate data from Infor BaanIVc4 company 100 to Infor LN company 200.
- *"UNIC"* code is reserved for the Unicode correction feature. This feature can replace illegal String characters in LN Strings. For example, when migrating to a Unicode environment it is not allowed to use Characters greater then character number 127. The charter "Ü" is not allowed and can be replaced by "UE".

The Maintain Values by Conversion Codes (bcgct991 table) can be specified in these ways:

- Manually add the old value and the new value
- Retrieve the Old value list using the master data table at the source and fill the new values manually.
- Create a function to generate the new value based upon the old value.

Do not specify all the source codes and a new value for it, but only for the codes to be replaced. If for a conversion code the old value is not specified the function returns the unchanged master data value (old value). If for a conversion code the old value is specified but the new value is empty the function returns the unchanged master data value (old value).

Unicode Correction

The BaanIV source environment is a non-unicode environment where the standard installation of LN10.5 and higher are Unicode installations. This change can lead to migration issues when in the source characters are used which are not allowed in the target environment. During a full migration we do not want to scan all strings for illegal characters as this is bad for performance. First a migration to the target environment is done without a correction. At the target environment, the migrated data is checked for which table fields illegal characters are used. For the list of fields, the migration logic pack needs to be adjusted so the migrated string is checked for not allowed characters. To make the modification the recoding utility can be used with the recoding code "UNIC". If this recoding code is used and the correct table field is entered the utility makes the required changes in the customized logic pack automatically. After rebuilding the Logic Pack the Unicode correction function is used to correct the string fields during the dumping process. You can define in

the “UNIC” recoding values which characters must be checked and how these characters needs to be replaced.

UTC conversion

In the BaanIV source environments dates cannot be stored as UTC dates as is possible in the LN10.x target environment. During the dumping process if the target field is an UTC date the BaanIV date is converted into an UTC date.
Until LN10.5 the maximum allowed UTC date was 19-janurary-2038. For BaanIV dates after 2038 the date was converted to the maximum allowed UTC value.
From LN10.5 you can have UTC dates up to 30-december-9999. When the target version can handle these date you can set the “Is Target UTC40 format?” (bcgct000.ituf) Migration dump parameter to “Yes” and then the date is converted in UTC dates with a maximum of 30-12-9999.
For dates beyond 19-january-2038 the conversion is done without accounting for the DST time.

Appendix A Validate Data Integrity

Purpose of the bdvalidate utility

The bdbvalidate utility analyzes data in Baan environment to prevent data corruption. Data is corrupted when it does not comply with the data definitions of the corresponding Baan environment. This utility is a part of the tools porting set.

It is very important to validate the data either in the source environment before actual migration or in the target environment after the migration or both. The goal after migration is to get the source data transferred to the target side in a correct state which the target side expects. Not solving the Data validation errors at the LN side can result in program failures or even prevent you from upgrading within the target LN environment.

It is required to validate the data at the LN side and optionally at the BaanIV side.

When validating data, the reorganize table session with option data and indices is used. This is a slow process, because it first dumps the data and then reloads it (bdbpre & bdbpost). During the reload process, the data is verified against the data dictionary. When an error occurs in the data integrity, the rest of the data in that table is skipped. This process is time consuming and not accepted by customers.

The bdbvalidate utility is a solution for the problem discussed earlier. The utility uses the same process to check the data, but does not dump and reload the data. This process is almost 10 times faster than the previous method. The utility enables you to identify the exact record wherein the data is incorrect (by the primary index) and specifies the type of error.

Reporting Sessions

In the Infor BaanIVc4 Engine environment, the **Check Data Integrity** (bccnv4233m000) session provides options to carry out the **bdvalidate** utility and prints a report of the erroneous data.

Bdbvalidate6.x Usage

The usage info: (options look similar to the ones for bdbpost, bdbpre and bdbreconfig)

```
Usage: bdbvalidate6.2 [options] -N <tablename> -C <compnr>
```

```
Options:

-C <compnr>      Validate tables in companies <compnr>. <compnr> may be:

1. List of one or more company numbers: -C 112 113 ...

2. Range of company numbers: -C 112-114

-E <file>        Redirect error output to <file>

-I <file>        Validate tables listed in file. Each line in the file must
contain exactly one table name.

-N <table names> Validate tables in list <table names>. For example: -N
dbtst100 dbtst120

-p <pack. comb>  Use package combination <pack. comb>

-R <file>        Create report file

-s               Suppress messages

-u/-U            Print usage information

-v/-V            Print version information

-z <number>      Zap to the next table after <number> errors
```

Types of domain constraints

These domain constraints are checked by this utility:

- F: Filled constraint
- R: Range expression constraint
- C: Uppercase / lowercase constraint
- A: Left / center / right alignment constraint
- L: Legal / Illegal characters constraint
- D: Magnitude of floats and doubles

Filled constraint

```
:FILL:1
```

A Filled constraint is violated in these situations:

- A numeric value equals zero (0)
- A (multibyte) string value equals a string containing spaces only
- A raw string contains zero (0) bytes only

Range expression constraint

```
:RANG:$$ IN [1,100]
:RANG:$$ IN "^[A-Za-z_][A-Za-z][0-9]*$"
:RANG:strip($$) IN "^[A-Za-z]*$"
:RANG:$$>1 AND $$<10
:RANG:$$ in "[a-ln-np-qs-z][a-z]" or $$ = "*" or $$ = "* "
```

A Range expression constraint is violated if the evaluation of the expression, where the symbol "$$" is substituted by the value, yields False (0).

Uppercase / lowercase constraint

```
:CONV:u
:CONV:l
```

An Uppercase constraint is violated if the string value contains a lower case character.

A Lowercase constraint is violated if the string value contains an upper case character.

Left / center / right alignment constraint

```
:ADJU:l
:ADJU:c
:ADJU:r
```

Legal / Illegal characters constraint

```
:LECH:1234567890
:LECH:-:;,.\|/
```

Legal characters constraint is violated if the string value contains any character not included in the legal characters.

Note: a space (' ') is always implicitly added to the legal characters, if a Legal character constraint is defined.

:ILCH:_&:

:ILCH:*/

:ILCH:/\$&~<>,;:"{}|`'^*()#@[]?

The **Illegal characters** constraint is violated if the string value contains a character that is included in the illegal characters.

Magnitude of floats and doubles

:DIGV:3

A **digv** constraint is violated if the float or double value in decimal notation contains more than the specified number of digits before the decimal point. For example, if the **digv** constraint specifies 3 digits, then the value 1234.56 violates the constraint, whereas the value 123.45 does not violate the constraint.

Note: only positive numbers are checked against a digv constraint.

Checked constraints

This table displays the constraints that are checked for each database type listed in the leftmost cell. If a cell is ✓, the constraint is checked for all the database types. If a cell is empty or has a (×),the constraint is **not** checked for all the database types:

	Fill (F)	Range (R)	Case (C)	Alignment (A)	Legal chars5 & Illegal chars (L)	Digv (D)	Diga
BDB_TEXT 0 BDB_MAIL 0 BDB_BITSET 0	×	×					
BDB_CHAR BDB_ENUM 2 BDB_SHORT BDB_LONG	✓	✓				×1	
BDB_FLOAT BDB_DOUBLE	✓	✓				✓4	×3
BDB_STRING BDB_TSS_STRING	✓	✓	✓	×6	✓5		
BDB_RAW	✓						
BDB_DATE	✓						

BDB_TIME	✓

Notations:

- [0)] No constraints are checked for the types BDB_TEXT, BDB_MAIL and BDB_BITSET.
- [1)] Number of digits before the decimal point (digv) is ***not*** checked for the BDB_CHAR, BDB_ENUM, BDB_SHORT and BDB_LONG types.
- [2)] The enum values are ***not*** checked for the BDB_ENUM type. So, if there is no constraint defined for a column of type BDB_ENUM, ***any*** value is permitted.
- [3)] The number of digits after the decimal point (diga) is ***not*** checked for the BDB_FLOAT and BDB_DOUBLE types.
- [4)] Negative numbers are not checked against the digv constraint for the BDB_FLOAT and BDB_DOUBLE types.
- [5)] If the Legal Characters constraint is defined, the space (' ') is always included in the legal characters.
- [6)] The Alignment constraint is ***not*** checked for BDB_STRING and BDB_TSS_STRING types.

Appendix B DCA 3.x Differences

This section gives a short overview of the enhanced features and fixes compared with the previous version of the DCA Conversion Adapter3.x BaanIV Engine

If the previous version is DCA3.1 or earlier see all changes in section DCA3.2 and DCA33.

If the previous version is DCA3.3 then see the changes in section DCA34.

DCA3.2

Enhancements DCA32

Synchronized dump and Load (LNUI)

The LN engine is now fully LNUI enabled. As from the target environment it is no longer allowed to start another environment process the synchronized dump and load process is changed. Dumping data and loading data at the same time is still possible but you must start the dump engine and then start the load engine.

UTC40

The dump engine can now also convert dates in UTC dates beyond 19-january-2038.

Database support

The DCA3.4 baanIV dump engine now also can handle the migration to an EDB-Postgress as a target database.

String Conversion functions

New conversion function added in bcrunconvdll for handling string conversions. Each string is now checked if the string is filled. If the migrated string is empty, the string is migrated as a single space string for Mssql and Oracle target databases. For EDB as a target database the migrated field is an empty string in this case. For the other databases the string field gets the full string length. For Mssql, Oracle en EDB target databases the size of the data dump files is therefore reduced.

Fixes DCA32

UTC conversion improvements

Wrong maximum utc date was returned for American time zones.
The UTC() function in bcrunconvdll is redesigned for the implementation of the UTC40 change.

Error during simulation from the DCA Studio

For some migration mapping objects the simulation output remained blank after starting the simulation twice.

DCA3.3

Enhancements DCA33

Verify in debug mode

From the DCA Studio you can verify a mapping object in debug mode via a Specific menu option. It is not required to use the DCA admin user to compile a mapping object in debug mode.

Fixes DCA33

Programming Constructions

In some migration components programming constructions where used which are not allowed in older porting set versions. These constructions are replaced by others with the same result.

UTC Dates and Time zones

The time zone of the user was not always correct used within the UTC conversion.