Data Conversion Adapter Studio User Guide

Data Conversion Adapter 3.4 Suite

Copyright © 2024 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: Data Conversion Adapter Studio 3.4

Publication date: December 12, 2024

Document code: U9093F

Contents

About this guide **7**
Intended audience 7
Related documents 8
Contacting Infor 9

Chapter 1 Introduction **10**
Purpose 10
Scope 10
Limitations 10
Definitions, acronyms, and abbreviations 11

Chapter 2 Overview **13**
Data Conversion Adapter 3.4 Suite 13
Data Migration Solution 14
Purpose of the DCA34 Studio 16

Chapter 3 Starting with the DCA Studio **18**
Installation of DCA34 Studio 18
Create a DCA34 Studio User 19
Log on to DCA Studio 20

Chapter 4 Master Data **22**
Users 22
Applications 24
Define an application 25
To Define a Non Baan Application 25
To define a Baan application 28
Migration profile 28
To define a migration profile 29
Function repository 30
How to define a function repository? 30
Recoding data 31

To define a recoding code 32
History & Multiple Process 32
To set up History & Multiple Processes 33

Chapter 5 Conversion Rules Modeler 34

Maintain conversion objects 34
Special Features 35
To create a new mapping object 36
Define conversion rules 37
Special use cases 42
List of Special keywords/functions for DCA Studio Mapping Objects 43
Conversion Build 43
To create a build 46
Search 46

Chapter 6 Utilities Data Comparison 47

Compare DD's 47
DDC Results 48
Show DDC Result 48
Fine-Tune DDC Result 50
Compare Migration Profiles 51
MPC Results 52

Chapter 7 Utilities - Miscellaneous 53

Where Used and Not used functionality 53
Generate Default Mappings 53
Upload Field Descriptions 54
Export Mapping Rules 55
Import Mapping Rules 56
Export Rules Patch 56
Import and apply patch 57

Chapter 8 Coversion Documentation 58

Conversion topics 58
Upload Document 58
Export object documentation 59

Chapter 9 Migration Process Flow 60

Install 61
Install DCA Studio and check for updates 61

Import 61
Import Logic Pack Rules (base) 61
Import Logic Pack Rules (update) 61
Applications 62
Define and load source application 62
Define and load Target application 62
Profiles 63
Define customized migration profile 63
Assign users to the profile 63
Mappings 63
Define conversion rules 63
Verify 64
Verify conversion rules 64
Simulate conversion rules 64
Build 65
Create a customized Object Pack 65
Create a customized Rules Pack 65

Chapter 10 DCA Server Utilities 66

DCA Legacy Engine 66
DB Config Utility 66
Load DataDictionary 67
Recoding 68
Rules Recovery 72
Script Debugger 72
User Manager 75

Appendix A Creating an ODBC Connection 77

ODBC setup for Access Database 77
ODBC setup for Legacy Database 78

Appendix B Lookup Database 81

Purpose 81
To define a lookup database 81
To access data in lookup database 81

Appendix C Troubleshooting 89

Errors using the DCA Studio 89

Appendix D DCA 3.x Differences 91

Enhancements DCA3.2 91
Fixes DCA3.2 93
Enhancements DCA3.3 95
Fixes DCA3.3 96
Enhancements DCA3.4 97
Fixes DCA3.4 98

About this guide

This document is a user guide that describes how to work with Data Conversion Adapter 3.4 Studio designed for data migration. This document describes the task-oriented approach to work with Data Conversion Adapter 3.4 and describes the important fields and their actions.

This document is intended for Data Conversion Adapter users. To use this document, this basic knowledge is required:

- Baan data conversion processes
- Knowledge of web browsers, such as Internet Explorer 11.0 (or later) Edge or Chrome.

Intended audience

This document is intended for the Data Conversion Adapter users who migrate their existing Baan or legacy products to latest releases of Infor products.

This document is intended for the DCA34 Studio users such as: developers, and migration consultants. Users who are maintaining or creating Logic Pack Rules with the Data Conversion Adapter 3.4 Studio. Knowledge of Infor BaanERP, Infor LN and/or legacy environment is required. Technical and functional knowledge of Baan or legacy application helps you more to perform the tasks. The user must have basic knowledge of the Microsoft Windows systems, Microsoft SQL Server 2022.

The DCA Studio is only intended for Infor Migration Consultants, Infor Migration Logic Pack Developers and Infor Migration partners who are approved for migration. An approval can be retrieved via Infor Solution Management. It is not allowed to install the DCA Studio at a migration customer system.

Related documents

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

These references served as inputs to the Installation Guide:

Infor Solution License Manager 7.5 Installation and Configuration Guide (U9200G US)

Other Data Conversion Adapter 3.4 documents are:

https://inforsaas.service-now.com/kb?id=kb_article_view&sysparm_article=KB2274580

Data Conversion Adapter 3.4 - Infor LN Engine - User Guide *U9130G US*(LNEngine_UserGuide)

Data Conversion Adapter 3.4 - Infor LN Engine - Installation Guide *U9131G US*(LNEngine_InstallationGuide)

Data Conversion Adapter 3.4 - Infor B50c Engine – User Guide *U9137F US*

(B50cEngine_UserGuide)

Data Conversion Adapter 3.4 – Infor B50c Engine – Installation Guide *U9138F US*

(B50cEngine_InstallationGuide)

Data Conversion Adapter 3.4 - Infor BaanIVc4 Engine – User Guide *U9149F US*

(BaanIVEngine_UserGuide)

Data Conversion Adapter 3.4 – Infor BaanIVc4 Engine – Installation Guide *U9150F US*

(BaanIVEngine_InstalltionGuide)

Data Conversion Adapter 3.4 – Legacy Engine – Installation Guide *U9096G US*

(LegacyEngine_InstallationGuide)

Data Conversion Adapter 3.4 – Data Conversion Adapter Studio – User Guide *U9093F US*

(DCAStudio_UserGuide)

Data Conversion Adapter 3.4 – Data Conversion Adapter Studio – Installation Guide *U9094F US*

(DCAStudio_InstallationGuide)

Data Conversion Adapter 3.4 – Data Conversion Guide for Baan50c to LN *U8920F US*

Contacting Infor

If you have questions about Infor products, go to Infor Concierge at https://concierge.infor.com/.

If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site 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 and purpose of the *DCA34 Studio User Guide*.

Purpose

The purpose of this document is to describe the functional handling of the Data Conversion Adapter 3.4 (DCA34) Studio.

Scope

This document focuses primarily on the DCA Studio aspect of Data Conversion Adapter 3.4 suite solution. This document serves as a User Guide for DCA 3.4 Studio.

Limitations

The known limitations of DCA Studio 3.4 when the source application is non-Baan:

- Application table names cannot be more than 60 characters.
- Spaces are not allowed in the table names. For example, an application table such as **Order Detail** is not supported.
- Binary data fields are not supported. Currently you can model binary data fields in the DCA Studio. However, the Engine fails to execute conversion for these tables during execution. In future, the support for binary fields can be enabled in Engine.

Definitions, acronyms, and abbreviations

Acronyms	Definitions
Application	An application in DCA Studio stores the data model pertaining to that application.
Conversion Object	DCA Studio uses pull strategy for data migration. In the Studio, you select a target application table and indicate the method to populate the data from source application tables. This is called defining conversion rules. A target table for which conversion rules are defined is called a Conversion Object. Therefore, the Conversion Object and target table are synonymous.
DCA	Data Conversion Adapter suite. It is a set of tools to assist you migrating data from a legacy ERP or BaanERP environment to an Infor LN environment.
DCA34	Data Conversion Adapter 3.4.
Dcaadmin	This refers to the DCA user which has the “Admin Role” within the DCA Studio.
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 the following 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, Logic Pack of Infor BaanIVc4SP15 to Infor LN 10.4. This logic pack consists of the conversion components to Migrate data from BaanIVc4 Service Pack 15 to the Infor LN 10.4 release.
DCA Object Pack	The (DCA) Object Pack is the collection of conversion rules programs as executables and can be installed at the source environment. They are used by the dump 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 conversion rules for a specific point- to-point data migration defined in the DCA Studio. The (DCA) Rules Pack consists of the following 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. The (DCA) Object Pack is used by the source environment engines 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 created by the DCA

	Studio are used by the source environment Engines to do the actual data extract.
DSN	Data Source Name.
ERP	Enterprise Resource Planning.
IE	Internet Explorer.
IIS	Internet Information Services.
Lookup database	An additional database required for complex data migration process, also called the pre-conversion database, in which the user can define default values, mapping values, and so on.
MDAC	Microsoft Data Access Control.
ODBC	Open Database Connectivity.
OLEDB	Object Linking and Embedding for Databases.
RDD	Infor ERP products store the data models in a set of files called Runtime Data Dictionary (RDD).
Server License	A license type in which a product license is linked to specific servers.
SLM	Solution License Manager: A central application that provides licenses to the Infor products. The SLM handles the license requests for Infor applications or partner applications bundled with Infor products.
SLM Server	The central component of SLM that handles the licenses requests of the adopted applications. In a client installation, a SLM server handles the licensing requests from various Infor products.
SLM Client	The component installed with the application to handle licensing with the central SLM server.
Target Application	Infor application to which data migrates.
XML	Extensible Mark-up Language.

Chapter 2 Overview

Data Conversion Adapter 3.4 Suite

The Data Conversion Adapter 3.4 enables data migration from any legacy systems to any Infor LN system or an Infor BaanERP system to any Infor LN system.

The Conversion Adapter Suite consists of these components:

- Data Conversion Adapter Studio (DCA Studio)

  You can use the DCA Studio to model and maintain migration rules packs. In a migration rules pack you can maintain migration mapping rules between a source and a target Infor ERP application.

  Conversion rules are developed for a point-to-point data migration. For example, a data migration from Infor BaanERP BaanIVc4 to Infor LN. The rules can be exported from DCA Studio for distribution, and then they are called a **Rules pack** for that migration.

  For migration of legacy systems to Infor LN, you must use VB Scripting to maintain rules. The rules are exported in the form of XML files, which you can run to perform the final data conversion, using DCA Legacy Engine.

  You must use Baan4GL scripting for the mapping rules to migrate data from one release of Infor BaanERP to Infor LN. The studio uses the mapping rules to generate conversion scripts, and compiles it against the source Infor ERP server. The respective dump engine is used to execute these compiled objects.

  The DCA Studio is only intended for Infor Migration Consultants, Infor Migration Logic Pack Developers and Infor Migration partners who are approved for migration. An approval can be retrieved via Infor Solution Management. It is not allowed to install the DCA Studio at a migration customer system.

- Data Conversion Adapter Legacy Engine (DCA Legacy Engine)

  The DCA Legacy Engine is formally known as the DCA Scheduler or DCA Engine. This DCA Dump engine is used to perform data migration of legacy systems to an Infor LN system. The input is the DCA Studio generated XML files from a Legacy Logic Pack.

- Data Conversion Adapter Dump Engine (DCA Dump Engine)

  You can use this component if you migrate data from an earlier version of Infor BaanERP/LN to a new version of Infor LN, for example, BaanIVc4 sp15 to Infor LN. The DCA dump engine consists of a set of sessions developed in Baan4GL, and grouped into pre conversion processes and actual data extraction processes and is installed in the source environment. The input for this component is the compiled conversion objects generated by DCA Studio.

  These Dump Engines are available:

  - Data Conversion Adapter 3.4 B50c Engine

- Data Conversion Adapter 3.4 BaanIVc4 Engine
- Data Conversion Adapter 3.4 LN Engine

- Data Conversion Adapter Load Engine (DCA Load Engine)

  You can use the DCA load engine only if you migrate data from an earlier version of Infor BaanERP, Infor LN, or from any other legacy system, to a new version of Infor LN. For example, Infor LN. The component consists of a set of sessions developed in Baan4GL scripting, and grouped into actual data loading processes and post-conversion processes. The input to this component is the data files generated by the DCA dump engines.

  ```
  Data Conversion Adapter 3.4 LN Engine
  ```

- Data Conversion Adapter Rules Packs (DCA Rules Pack)

  These rules packs are the real conversion rules developed using the DCA Studio. Conversion rules are specifically developed for a point-to-point data migration. For example, migration of data from Infor BaanERP BaanIVc4 to Infor LN. The rules can be exported from DCA Studio for distribution, and then they are called a **Rules pack** for that migration.

- Data Conversion Logic Pack

  A data conversion logic pack consists of the following components:

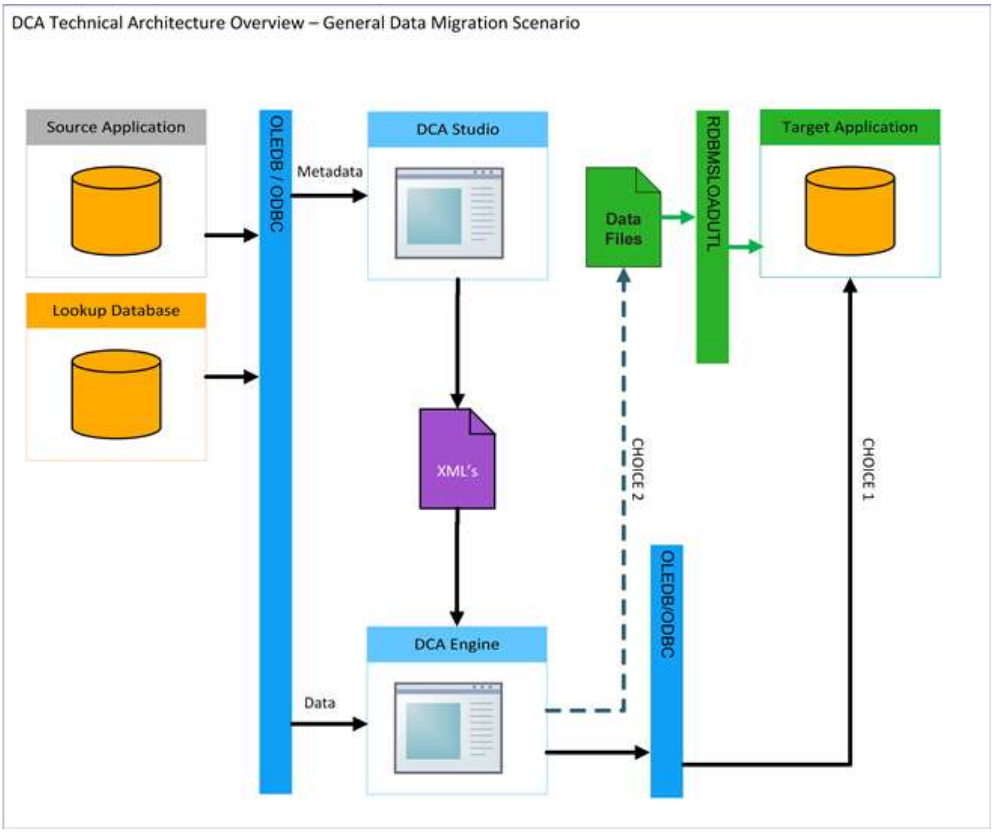
  - Data conversion adapter Rules Pack for the specific source to target application.
  - A set of Pre migration process sessions
  - A set of Post migration process sessions.

Some examples of existing data conversion logic packs:

- Infor BaanERP B50c sp17 to Infor LN Logic pack.
- Infor BaanERP BaanIVc4 sp15 to Infor LN Logic pack.

Data Migration Solution

The DCA Data Migration Solutions for any ODBC supported legacy application to Infor products is shown in this diagram:



The DCA 3.4 accesses source and target databases, using ODBC or OLE DB providers. Therefore, you must install the providers for the source and target databases. The installation process for the data providers is not part of this document.

Lookup Database

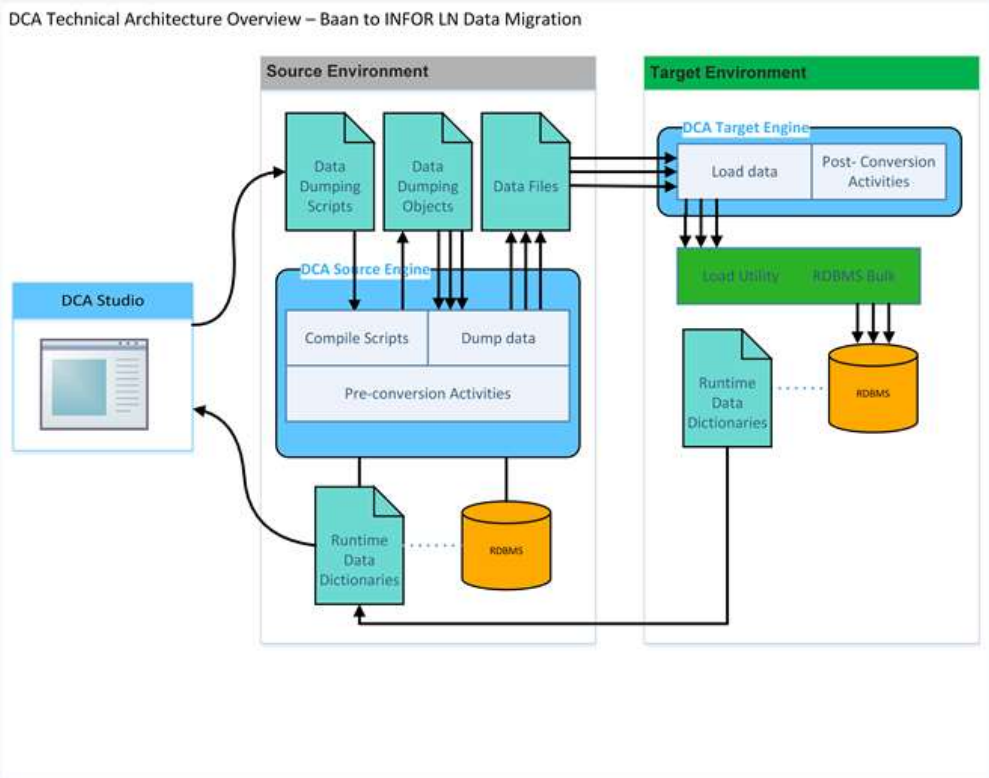
Lookup DB is an optional component in the DCA data conversion solution. For any complex data migration, you must have an additional data store other than the source database, do the following:

- Prepare manual data that you cannot retrieve from your source application.
- Generate additional data from your source application as a part of pre-conversion preparation.
- Recode some of your master data.

For these requirements, you can define your own table definitions in any of the ODBC/OLE DB supported data store. To access this data store easily, you can simply use the DCA-provided functions or you can code specialized functions in DCA Studio.

Note: You must use the look-up database in case you cannot define any tables as part of your source application. You need not use look-up databases when you can create pre-migration tables as a part of your source application.

The DCA Data Migration Solutions for Baan products to Infor LN 10.x is shown in this diagram:



You can carry out the complete conversion process in one of these ways:

- Dump data in the source environment and, in parallel, start to load to LN target Environment. The user can start multiple processes for dumping and loading, to improve the overall performance. This is normally referred as **dump and load.**
- Dump data in the source environment, later transfer the data files to the target environment and load the files into the target database. The user can also invoke multiple processes for dumping, and for loading, to improve the overall performance. This is normally referred as **dump and load later**.

This document focuses on migration of data from legacy systems to Infor LN products, and from Baan to Infor LN 10.x. It describes also the detailed processes you must run in the DCA Studio, DCA Legacy Engine, DCA Dump Engines, and DCA LN engine.

Purpose of the DCA34 Studio

Data Conversion Adapter Studio is developed to make the data migration process easy and more flexible across the systems.

To simplify point-to-point data conversions process, the concept of Data Conversion Adapter is evolved.

The product covers all these phases:

- Importing data models of various products
- Definition of data conversion rules, for a specific data conversion scenario

- Automatic generation of conversion scripts from conversion rules
- Compilation and running of the conversion scripts
- Actual data migration

The Data Conversion Adapter is mainly divided into these modules:

- Data Conversion Adapter Studio: Models a specific point- to- point data migration scenario. Studio generates conversion objects that are used by the Engines to perform the actual data migration. DCA Studio is web-server component, which can be installed in Microsoft's Internet Information Services (IIS).. Microsoft Internet Explorer (IE) is not supported in the latest Windows version hence recommended to use Microsoft Edge as a default browser to access DCA studio. With the latest update of Chrome browser, this extension is no longer free and is available on free trail period. Hence it is recommended to use Microsoft Edge as a default browser.
- Data Conversion Adapter Engine: It is a standalone MS-Windows executable that performs the actual data conversion process. In case you migrate data from Baan to Infor LN, the Engines are available as a part of Baan release.

  Currently these Engines are available:

  - DCA34 BaanERP BaanIVc4 Engine
  - DCA34 BaanERP B5.0c Engine
  - DCA34 LN Engine.
  - DCA34 Legacy Engine

Chapter 3 Starting with the DCA Studio

This chapter describes the prerequisites to start working with the DCA3.4 Studio website.

Installation of DCA34 Studio

The DCA Studio is only intended for Infor Migration Consultants, Infor Migration Logic Pack Developers and Infor Migration partners who are approved for migration. An approval can be retrieved via Infor Solution Management. It is not allowed to install the DCA Studio at a migration customer system.

You can install DCA34 Studio on either a Microsoft Windows 2016 (64bit) Server Enterprise Edition or a Microsoft Windows10 / Windows 7 Enterprise (64bit) operating system. For details on the installation of the DCA34 studio including hardware and software requirements see the *Data Conversion Adapter 3.4 Studio – Installation Guide (U9094F US)*.

The Studio requires an Internet Information Server 7.5(IIS7.5) Web server, and a Microsoft SQL Server 2024 database server.

DCA Studio is licensed using the Solution License Manager (SLM). DCA Studio's product ID is 1000 and uses a Server License type. To license the DCA Studio, complete the following steps:

1. Install SLM Client 7.5.0.6 on the DCA34 Studio server machine. If the SLM is installed on another server validate the connection.
2. Configure your SLM server for the product ID ***1000***. The license type is ***Server***
3. Contact the Infor Validation group (E-Mail: Infor.Validation@infor.com) to obtain the activation key from Infor Validation group

Note: For more details on software licensing using SLM, please see the *Infor Solution License Manager 7.5 Installation and Configuration Guide (U9200G US)*.
It is not allowed to use the temporary license key with the DCA3.4 Studio.

To access the DCA34 studio website, Microsoft Internet Explorer (IE) is not supported in the latest Windows version and hence the default and supported browser is Microsoft Edge which can be used in internet explorer mode.. With the latest update of Chrome browser, this extension is no longer free and is available on free trail period. Hence it is recommended to use Microsoft Edge as a default browser. See the installation guide for specific settings.

Create a DCA34 Studio User

This section describes how the user can logon and start using the DCA34 Studio. Before you start using the DCA Studio make sure the IE settings and other prerequisites are handled.

The system administrator creates the users who can access DCA Studio. Users can be created with the DCA User Manager which comes with the installation of the DCA34 Studio. One of the created users is assigned to the DCA admin role. In previous versions, it was required to create the user “baan” which was the admin user but this is no longer required. Any created user can have the admin role, but there can only be one user with the admin role.

To create a new user:

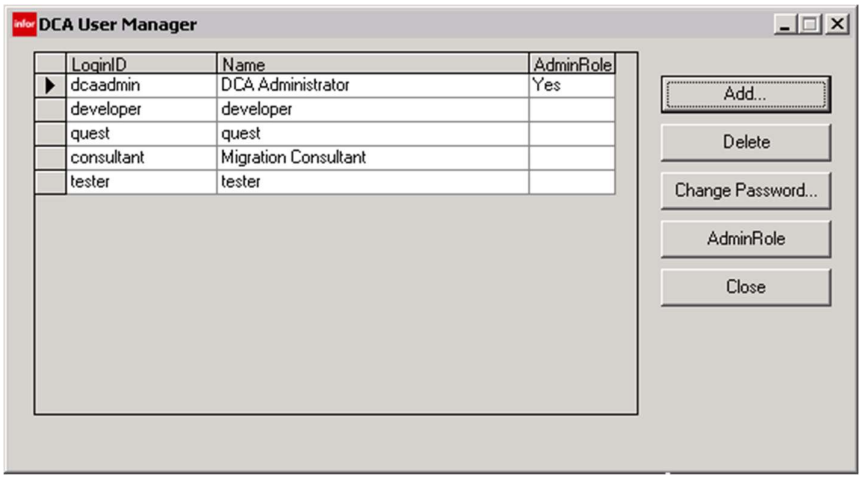
1 Click **Start → All Programs → Data Conversion Adapter → User Manager**. The DCA User Manager login window is displayed.

The following table describes the fields and commands of DCA User Manager login screen:

Fields	Description
Database Server	This is a non-editable field that displays the database server name used by the DCA Studio.
Database	This field is a non-editable field that shows the SQL Server database that is linked to the DCA Studio.
User Name	This field is a non-editable field that shows the login name used to connect to the DCA database. The user name is the name you have specified during the DCA Studio installation.
Password	Enter the password for the user specified in the User Name field. This user name and password is required to connect to the SQL Server database.
Commands	Description
OK	Connects to the DCA’s database and shows the DCA User Management window.
Cancel	Close the program without any action.

Note: To change the database server name, to create other DCA Database, and User name, see “DCA DB Config utility,” in Chapter 10, “DCA server utilities.”

2 Fill in the Password of the database server user and click **OK**. This will open the DCA User manager window. In this window, you can perform different tasks, such as **Add** a new user, **Delete** an existing user, and with **Change Password** the password of the selected user can be changed and or reset. With **AdminRole** to change the role of the selected user. Click **Close** to leave this screen.



Note: The length of the Login ID should not be more than 12 characters and the creation of the user '**dcaadmin'** is not required, but one user must have the Admin Role added.

Log on to DCA Studio

After the users are created using the DCA user manager you can login to the DCA Studio.

Connect from the DCA Studio server

1 Click **Start** → **All Programs** → **Data Conversion Adapter** → and start the Data Conversion Adapter.

The Microsoft Edgebrowser starts the DCA website with the login screen: http://localhost/DCA34/Login.asp. In the settings popup menu, click **Reload in Internet Explorer Mode** . For more information please refer DCA Studio Installation Guide

Connect from a Client machine

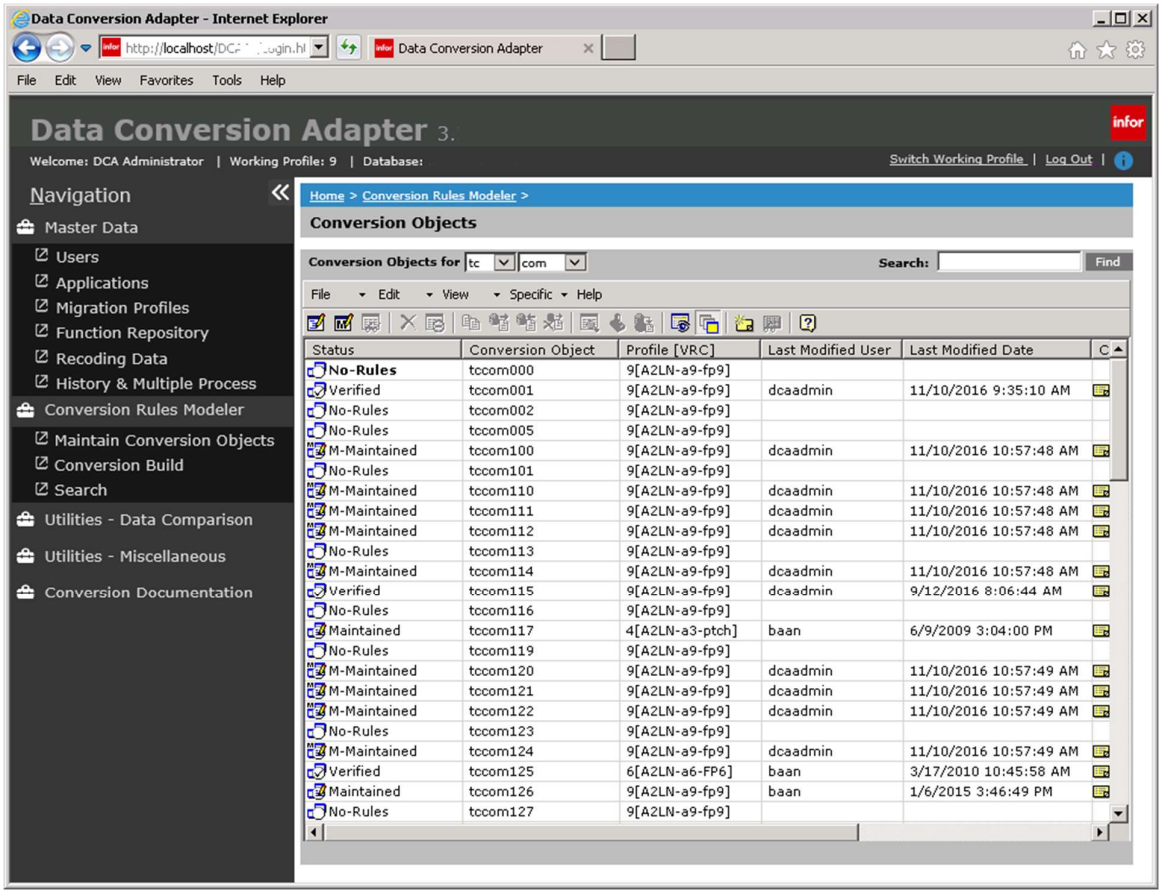
When connecting from a client machine start your browser and type this address:

`http://<DCA34 Studio Server name>/DCA34/Login.asp`

The browser starts with the DCA Studio login screen. Enter a valid user ID and Password and click **OK**.

The user can also change the password. Enter a valid user ID and Password and Click **Change Connection**. The administrator creates the user ID and the initial password for the respective users of this application. This screen provides an option to change the password. In the **Change Password** screen the user must enter the current password, the new password and must confirm the new password. By clicking **OK** the password is changed. By clicking **Cancel** the user leaves the **Change Password** screen.

Upon successful login, the **Data Conversion Adapter 3.4** Studio window is displayed:



The Data Conversion Adapter consists of the five modules, each of which is listed in the navigation pane. The following chapters in this document describe each of these modules in detail.

Besides the descriptions in the user guide there is also online help available contacted directly from the DCA34 Studio

Chapter 4 Master Data

This module consists of the master data entities that control the working of the DCA Studio. Details about each of these entities are provided later in this chapter. The Master Data module consists of these entities:

Users

Shows details about users who can access the DCA Studio application. Logged-in users can also modify their preferences here.

Applications

Used to store data models of a specific application.

Migration Profiles

Used to define a migration scenario; and groups all the migration rules for a specific data migration scenario. You must specify the source and target application here.

Function Repository

Used to define user-defined VB functions, mainly used while you migrate data from non-Baan applications.

Recoding data

Here, the user can maintain recoding master data codes. You use these codes mainly during real customer data migration, and not during Logic Pack development. The Recoding utility uses these codes. Recoding is the process by which you can standardize your master data codes during data migration to the new system.

History and Multiple processes

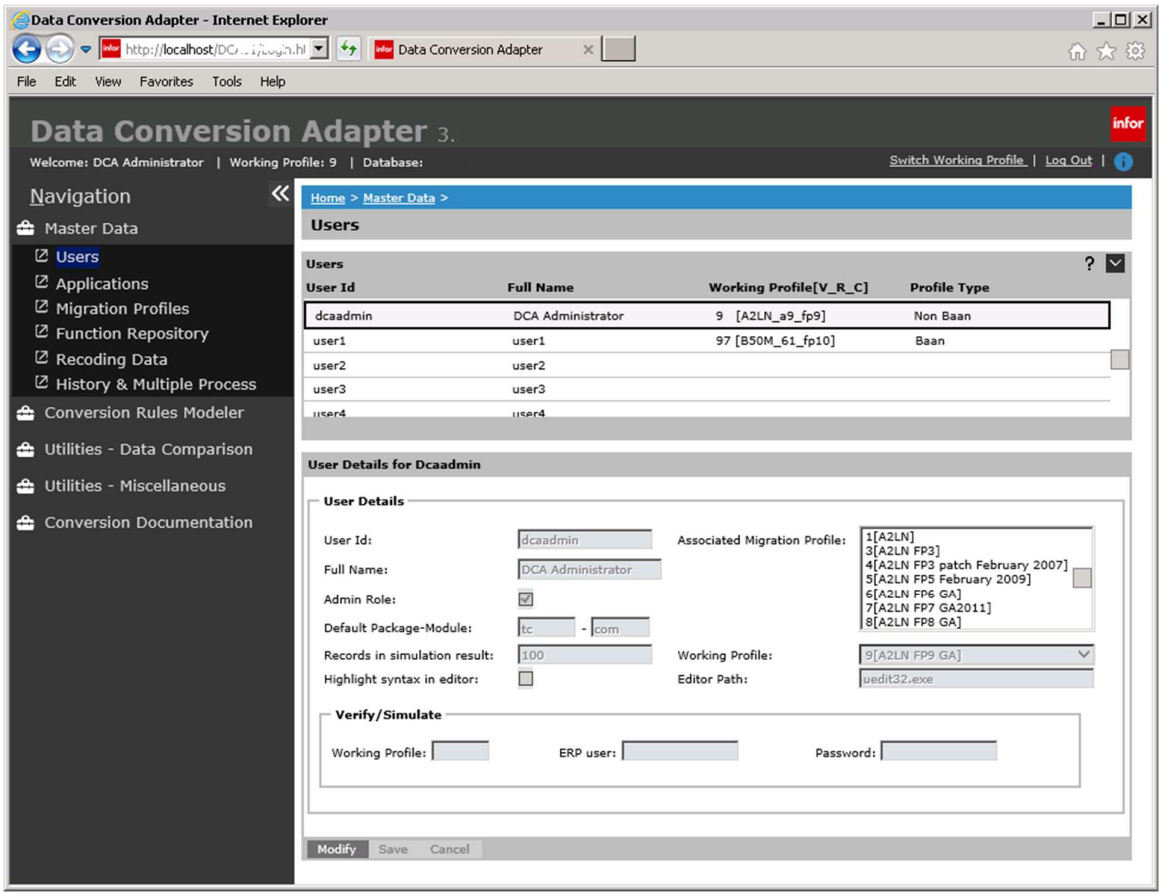
To migrate history data or to use multiple processes to extract data for a target application table, you must enter the source table field name by which we can identify history data or group source data to various data sets. This setup is mainly done when you perform the data migration for a specific customer. This information is used by Studio during the object pack creation, i.e., during conversion build.

Users

The Users page shows the users who are authorized to access DCA Studio. As a user, you might have the authorization to work with various profiles. However, you can only work with one profile at any given point of time. Click the **User ID** hyperlink to view the user details. You cannot

create any new users from this page. To create a new user, or to delete an existing user, or to reset the user password, see Chapter 3.

To access the User Management page, click **Master Data → User**. The **Users** window is displayed:



You can only change the settings for the user currently logged on to the DCA34 studio. You can view the settings of the other users.

Admin Role:

Login as the user who becomes the DCA administrator user and then click **Log on**.

After a successful login the user has to be assigned to an Administrator Role by checking the Admin Role option in the user Details Form:

Home →Master Data →Select your login User →Modify →Check Admin Role

There can be only one user assigned to the Administrator role and the Admin Role can only be added or removed by the logged on User itself.

For baan profiles the field ERP user will always be display a value even if the field is not set. If this field is not maintained by the user, the field displays a value according this flow.

1. User/Password is already populated [No action]
2. Get User/Password from Actual Profile
3. Get User/Password from Parent Profile

4 Get User/Password from Working Profile (BECS)

For a description of the fields on the **User** screen see the online help within the DCA34 studio.

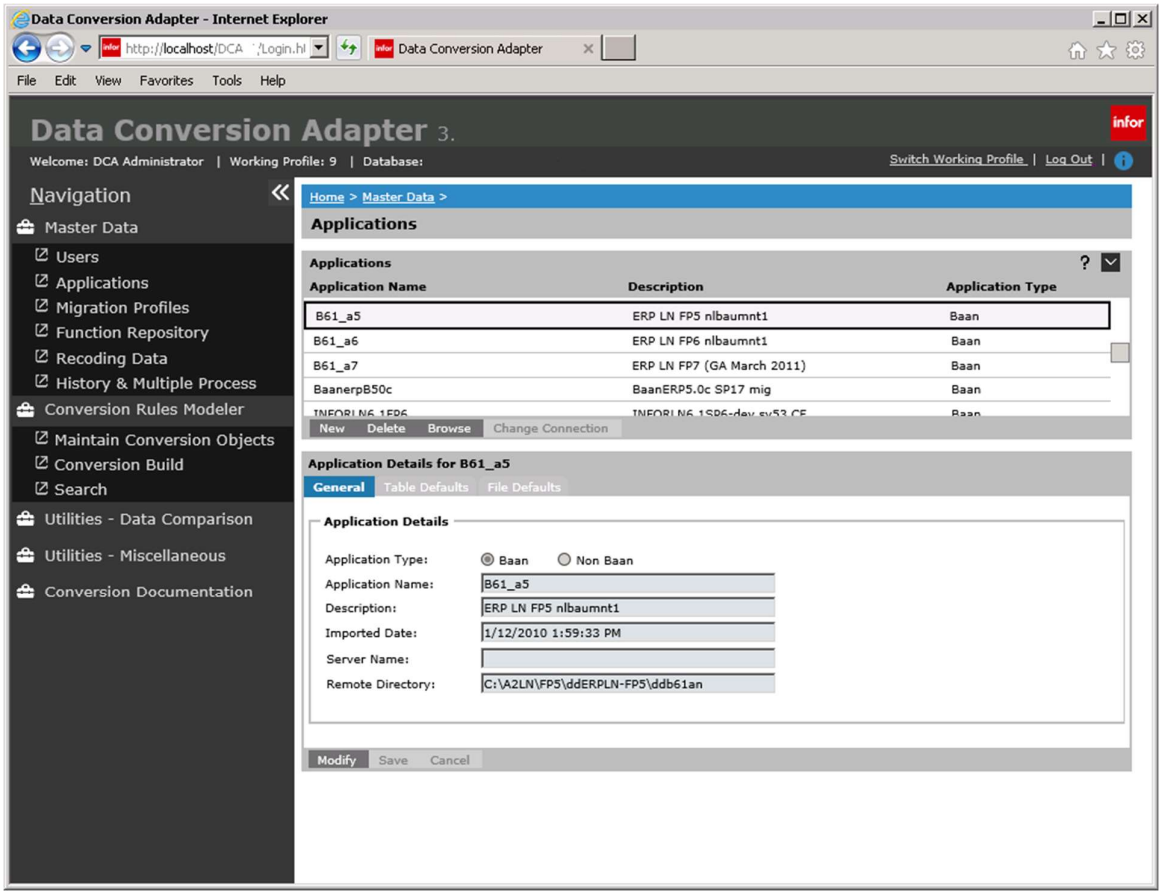
Applications

You can use applications to store data models of a specific application. In DCA Studio, you can define two types of applications:

- Baan Application.
  Enables you to store data models of a Baan application. Here, the data models are retrieved from the runtime data dictionary (RDD), which is a set to ASCII files. The DCA reads this data dictionary and stores the data models in the DCA repository.
- Non-Baan application.
  Enables you to store data models of applications other than Baan. Here, the data models are retrieved from the database directly by means of the ODBC/OLE DB provider for that database.

In the applications screen you can add, modify applications. With the browse function, you can view the loaded data definition for an application.

To access the User Management page, click **Master Data →Applications**. The **Applications** window is displayed. For a detailed description of the fields on the **Application** screen see the online help within the DCA34 studio.



General functions within the applications are:

- Create, delete and modify an application
- Reload the application content
- View application content

Define an application

As the process of setting up a Baan application differs from setting up a Non Baan application, both processes are explained.

To Define a Non Baan Application

To define a Non Baan application:

1 Click **Master Data** → **Applications** and click **New** to open the application screen.

2 In the application screen select the Non Baan application.

3 Enter an application name and description

4 For Non Baan applications specify on the **Table Defaults** tab the connection details to the database, from which data models needs to be imported. For details on how to create an ODBC connection see Appendix A. Test the connection to the database.

5 Click the **File Defaults** tab and configure the following to migrate data from plan ASCII files. Otherwise, ignore this tab. DCA uses Microsoft Text Driver to retrieve data from the files. The data files must have extension "**.txt**

6 Click **Save** on the general tab.

7 Select the application in the application overview page and click **Browse**.

8 Click **Import tables**. A new dialog screen will open to select which tables needed to be loaded in the application. Define the selection or choose to select all tables by marking the **Tables** checkbox. Click **OK** to start the import of the data definition into the application.

The new dialog screen will only contain the tables who are checked as *'In Scope'* in the Maintain Scope overview from the Staging Templates database

Special Use Cases:

In an application you can store tables from multiple databases. To load tables from multiple databases to one application in DCA Studio, complete the steps below:

1 Click Modify on the Application Details page.

2 Click the **Table Defaults** tab.

3 Change the connection details to the database and click **Save**.

4 Click **Browse** on the Application overview page, and select **Import Tables**.

5 Select the tables you want to import and click **OK**.

Special Use Cases:

Loading multiple Tables from an external MSSQL Database:

Reading from a MSSQL Database:

SQL-Server

1 Login into the server where MSSQL tables are located.

2 Start 'Microsoft SQL Server Management Studio'

3 Select Security->Logins

4 Right click on Logins

5 Select General

6 Click New Login

7 Select a page-> General

8 add a Login name: ***dca_reader***

9 Check: SQL Server authentication

10 Populate a password

11 Select the Default database in the dropdown.

12 Press Ok

13 Select a page-> Server Roles

14 Check one server role: ‘public’

15 Click Ok

Now the SQL Server is ready to serve data to the DCA.

ODBC Datasource connection

1 Start ODBC Data Source Administrator (32-bit) Tool on the DCA-server

2 Add a System Data Source entry in the System DSN Tab.

3 Click Add.

4 Select a ‘SQL Server Native Client’. Driver

5 Click Finish

6 Populate Name, Description and which SQL Server you want to connect.

7 Click Next.

8 Use SQL Server Authentication using a Login ID and Password .

9 Using a special user: ***dca_reader*** with corresponding password.

10 Click Next

11 Change the default database in the dropdown.

12 Click Next.

13 Click Finish.

14 Click Test Data Source.

Result should be contain “TEST COMPLETED SUCCESFULLY”

(Entered user/password combination will not be saved on your system!)

DCA-Studio

1 Start the DCA-Studio (DCA-admin user).

2 Select **Master Data → Applications**.

3 Select the Application Name

4 Click Change Connection

5 Update ‘New User’ to ***dca_reader*** with corresponding ‘New Password’.

6 Click Ok

Now generated scripts can connect to the MSSQL database in order to process the data.

To define a Baan application

1. Click **Master Data → Applications** and click **New** to open the application screen.
2. In the application screen select the Baan application.
3. Enter an application name and description and click **Save**.
4. A message providing an option to import RDDs from the Baan application server is displayed. Click **Next** to import the RDDs or select **Cancel**.
5. An XML Path can be added to enrich the already imported RDDs by reading additional XML Files. Click **Browse** to select the folder.
6. Enter the details in the **Baan Host Information** area in the Import Baan Application screen and click **Import**.

Note: When uploading the RDD/XML from a Local windows system the RDD/XML files must be present on the local system. In case the Infor ERP environment is on a different windows server the RDD/XML files must be copied to the local server.

Special Use Cases:

The following section describes scenarios with their specific solutions for Baan applications.

Case	Solution
Scenario 1: To reload the data models again in case of some changes in RDD. This situation generally occurs during the development of logic pack	For Baan Applications: Click Modify on the Application Details page. Click **Save**. DCA asks you if you want to import data models again. Click Yes.
Scenario 2: For a Baan Application, you can import tables selectively	Use the standalone application, present in the DCA server. To access this program, on the Start menu, click Programs → Data Conversion Adapter → Load Data Dictionary.

Migration profile

Migration profiles enable you to relate a source application to a target application that will define a data migration scenario. The users are linked to migration profiles and, for a given period, a user maintains conversion rules for a specific migration profile. Only the '**dcaadmin**' can create new migration profiles and assign users to the profiles.

Within the migration profiles you can create a dependency structure to other profiles. For developing new profiles, it is advised to use the option revision controlled. With this option all changes done on mapping objects for the profile are handled via a check-in/checkout mechanism.

To access the Migration Profile page, click **Master Data → Migration Profiles**. The **Migration Profiles** window is displayed. For a detailed description of the fields on the **Migration Profiles** screen see the online help within the DCA34 studio.

General functions within the migration profile:

- Create, delete and modify migration profile settings.
- Assign users to the profiles.
- Assign build settings for the profile.
- Check in all or undo checkout all mapping objects linked to the profile.
- Delete all revisions for the profile.

To define a migration profile

Only the '**dcaadmin**' can create profiles.

1 Click **Master Data** → **Migration Profiles** and click **New** to open a new migration profile details page.

2 Enter a profile number and a profile description.

3 Setup the source and the target application for the new profile

4 Enter a VRC for the profile

5 Set the derived from profile

6 Select if the new profile requires revision controlled or not. After revision controlled is turned on it cannot be turned off. But you can activate the revision controlled at a later stage. When revision controlled is selected also a development VRC is required.

7 Click **Save.** Now the migration profile is created.

8 Select the new profile in the profiles pane and Click **Users** to allow users to access the profile. The '**dcaadmin**' has always access to all profiles.

9 Click Settings to define the connection details for the profile. This is needed when migration objects for the profile are compiled or you want to create a build for the profile.

10 Click **Get Derived User** to retrieve the compiler settings for the users from the derived from structure.

Special use cases

This section provides several cases with specific solutions on the use of **Current VRC** and **Derived from Profile**. This section assumes you have created a migration profile for data migration from *Legacy* application to *Infor LN* and that the Version Release Customer (VRC) for this profile is *B61_a_st01*and the profile number is *85*. Consider the scenarios:

Case	Solution
You have created and released the conversion rules and want to perform maintenance activities for the rules in a new profile	Create a new profile with the same source and target application and, in the Derived from Profile field, select the original profile number, 85, which is frozen. Specify the current VRC as B61U_a_st02. The letter "U" in the version part is required as it indicates the new profile is an update profile.

You already have conversion rules for standard Legacy to Infor LN. You want to adopt these conversion rules for a customer, but the customer has customizations done on both Legacy-access database and on Infor LN.	You must define two new applications, such as A2LN_cust and I10EE_LN_cust, and upload the data models to respective applications. Define a new Profile; derive it from the standard Profile, 94. Enter the current VRC as B61_a_ABC. Assume the customer name is ABC Inc. Ltd.

Function repository

Function repository is useful if you are extracting data from non-Baan applications, that is, to write generic VB script functions, that can be used when building conversion logic. Functions are like ***includes***. To use a function in a conversion object, you must use this syntax:

"#include <function name>".

For example, *#include <common>*, where *common* is the function name. You must define this in the 'Variables/Includes' section of the conversion object.

To work with function repository, click **Master Data → Function Repository** and the Function Repository page opens. For a detailed description of the fields on the **Function Repository** screen see the online help within the DCA34 studio.

General functionalities within the function repository:

- Create and maintain functions/includes
- Revision controlled
- Copy repositories from other profiles
- Printing
- Adding comments to the function repository object.

How to define a function repository?

A function repository can only be created when the current working profile of the DCA user is linked to a Non-Baan application.

To define a function repository:

1. Open the function repository screen.
2. In the menu bar go to File and click **New**.
3. Enter a function name and description and click **OK**.
4. If the current working profile is set to "**Revision Controlled**" then before you can maintain the function repository it must be checked-out.
5. Mark the new created function repository and click **File** and **Open**. An editor screen opens where the user can program a generic (VB) function. **Save** and **close** the editor screen to return to the function repository screen.

6 If the current working profile is set to “**Revision Controlled**” then after maintaining the function repository it must be checked-in with a revision text.

Special use cases

Case	Solution
You have a function repository in a derived from profile but you must make a change.	It is only possible to maintain function repositories within your current working profile. Mark the required function repository in the derived from profile. Access the **specific** menu option and choose **copy to current profile**. The selected function repository is now also available for your current profile and you can start editing.
You made a change to a function repository for a revision controlled profile but you want to undo the change.	Select the function repository. Access **View** and choose **revisions**. A new screen will start with a list of all the revisions for this function. Select the revision to which you want to revert back and click **OK**. You will return to the function repository screen, the object is check-out and the situation of the selected revision is restored.
To see the added comment to the function repository object.	Select the function repository and access **View** and choose the option **comment**. This will allow you to view the comment. If you must maintain the comment access **Edit** and choose **comment**. You can also print the comment together with printing the content of the function repository. Access **File** and choose **Print**. Mark the option “**with documentation**”.

Recoding data

This feature enables you to recode master data, as a part of migration. The recoding functionality is available for the B50c Engine, BaanIVc4 Engine, LN Engine and the A2LN logic pack. In this section of the DCA Studio, you must maintain conversion code for each recoding scenario.

For example, to recode employee codes and warehouse codes, you must maintain two codes here, such as EMNO and CWAR. The codes can be defined by your own convenience. For each conversion code, you must maintain only one pair of old and new values. These values need not be the actual values, because the actual values are generated and maintained in the DCA’s dump engine or access database (A2LN).

To define a conversion code, click **Master Data** → **Recoding Data** to open the Recoding Data screen. For a detailed description of the fields on the **Recoding Data** screen see the online help within the DCA34 studio.

To define a recoding code

1 Open the recoding Data Screen

2 Click **New** and enter a recoding code and a description and click **Save**.

3 Enter an Old value, New value and click **Save**. For instance, old value = 0 and New value = 1.

4 The recoding code is created.

5 To apply the recoding code the Recoding Utility needs to be used.

Note: There are two reserved recoding codes:

- **COMP**:
  - To recode, the company numbers, that is copying one Infor BaanERP BaanIVc4 or Infor BaanERP B50c company to another company number in Infor LN, use the conversion code *COMP*, which is a reserved for company number recoding.
- **UNIC**:
  - This recoding code is used in the Unicode correction functionality.

You can enter dummy old value/new value pairs under conversion codes here, because you must enter actual old value/new value pairs under conversion code using the *Conversion Codes* session at the DCA's dump engine. However, you must create the same conversion codes (case sensitive) as the codes you create here.

The Recoding utility uses the conversion codes maintained here to recode the conversion rules. For more information on the Recoding utility, see "Recoding utility," in Chapter 10, "DCA server utilities."

History & Multiple Process

You can use this setting to complete actual data migration for a customer. The objectives of this utility:

- When source data tables have a huge amount of history or dead data,data that is not going to change, you can migrate the data in advance. During the live migration, the system down time is less. This use case is referred to as '***History data migration***'
- When some live source data tables have a huge amount of data, you can think of migrating the tables, using multiple processes. For example, a table can have 100000 rows and take 2 hours to migrate. You can decide to use two processes to migrate this table. The first 50000 rows are migrated by the 1st process and rest are migrated by the 2nd process. The total migration takes only 1 hour. This use case is referred to as '***Multiple processes by Conversion object***'

First you must identify which all conversion objects, you can migrate history data and which table requires more than one process to migrate data. This can be decided after the test migrations, executed in the customer's system. For each of these conversion objects, you must look the conversion rules and find out the source tables from which data is getting migrated. This can be easily achieved by looking at the `migrate` tag in the conversion rules, each `migrate` tag corresponds to a query by which you retrieve data from one more source tables. Now for each of this source table(s) you must identify the table fields using which you can identify history data, mostly you can find some date fields or status fields. To use multiple processes, you must divide

the complete data into set corresponds to number of processes to use. For this the best candidate is the primary key of that table.

Note: When you migrate history data, it is advised to use the primary key as the split criteria. If you are not using the primary key, then it is advised to use one process. If the 2GB file limitations is required and you want to use multiple processes makes sure each process does not exceed this limit.

The settings you do here are used during Conversion build and the scripts are generated accordingly. The actual boundary values for history/each of the processes are set in the respective Engines, against that conversion object.

Note: This setting is independent of any migration profiles. If you have any settings in this section, this information is read, whenever you do a Conversion build. To avoid this information, you must clear, 'Consider history & multi-process setup' in the build Settings.

To define this setup in Studio, go to **Master Data → History & Multiple Processes.** For a detailed description of the fields on the **History & Multiple Processes** screen see the online help within the DCA34 studio.

To set up History & Multiple Processes

To define a recoding code:

1. **Open** the History & Multiple Process Screen
2. Click **New** and enter a base conversion object and click **OK**.
3. Enter the "main query on" table and in the "Source field list" define the field name. Click **Save**.
4. Click **Verify** to validate the entered data.

Chapter 5 Conversion Rules Modeler

Use this module to maintain conversion rules, build the rules, *in other words*, create executable conversion scripts, and print reports that are useful during logic pack development or for logic pack rules customizations.

In the **Maintain Conversion Objects** the user can specify and maintain migration mapping rules.

If a user defines rules for a target table, the table is referred to as a conversion object. However, if a user defines conversion rules for a group of target tables, the table is referred to as a multiple conversion object, or M-Object. The M-Object defines conversion rules to migrate data from one or more source tables to two or more target tables.

The definition of M-object improves:

- Performance: The source table(s) are selected once and data is migrated to multiple target tables
- Maintainability: The data conversion logic specific to a group of target table is positioned at one place.

In the **Conversion Build** the user can create the migration rules object pack. The input for the conversion build is the conversion objects for a profile. The output is an executable version of the migration object which is used by the DCA dump engines.

In the **Search** section the user can do a search on the created migration profiles. With this functionality you can for instance search all profiles for a specific migration object or all objects in all profiles with a specific status.

Maintain conversion objects

The target application tables are organized under respective package and module. Here, the user can select a target table (package, module, and table) and maintain how data is migrated from the source application. The logic of data migration from source to target is called conversion rules and each of the target table is referred to as a conversion object.

The Conversion Objects page enables you to maintain conversion rules for the target tables in the migration profile. The tables are organized per ERP packages and modules. The conversion object and related actions appear in the Conversion Objects pane. To work with conversion object, click **Conversion Rules Modeler → Maintain Conversion Objects**. The Conversion Object screen is displayed. For a detailed description of the fields on the **Maintain Conversion Objects** screen see the online help within the DCA34 studio.

General functions within the mapping objects are:

- Create and maintain Mapping objects (Normal and M-scripts) and define mapping rules.

- Revision control via check-in / check-out mechanism, view revision comments.
- Copy mapping objects from other profiles.
- Copy mapping rules to other dumps within the same mapping object.
- Copy mapping rules to child profiles.
- Preview generated migration script.
- Verify and simulate.
- Printing.
- Adding comments to the mapping object and the Mapping rules. These comments are also used for creating migration Webhelp.

A mapping object can have these statuses:

- No-Rules:

  This is the initial status. No conversion rule is maintained for this table.
  In case the view is set to “All conversion objects” the status No-Rules means also the mapping object for the profile has no rules. But the same mapping object can have mapping rules for another profile.

- Comment-only:

  In case you maintain some documentation for the table without any conversion rules. For example, you specify the reason when the table is not migrated.

- Generated:

  The conversion rules are generated automatically from the Data Model comparison result using Utilities Miscellaneous → Generate Default Mappings.

- Maintained:

  Conversion rules are maintained for this target table. However, this does not guarantee that conversion rules are complete and correct.

- Verified:

  When you perform the verification, the conversion rules are checked for syntactical correctness. However, it doesn’t guarantee that conversion rules are complete and correct; you can still have run time errors.

- Error:

  The target table is set to this status when the verification of conversion rules fails. This indicates that, there are some syntactical problems in the conversion object.

- Expired:

  When an object is set to expired in the working profile. The expired objects are not considered during conversion build.

Special Features

Search a mapping object

There are two ways to search a mapping object:

- Enter a package and module and locate the mapping object in this package module.

- Enter the target table in the search area.

View toggle

Viewing the mapping object overview screen can be done in two ways:

- **All Conversion objects**: Enables you to view the conversion objects in the system.
  Note: It takes time to display the conversion objects, depending on the number of profiles you have in your DCA Studio.
- **View only latest conversion objects**: Enables you to view only the latest conversion objects going upward in the hierarchy. For example, if the same conversion object is present in both the parent and the current working profile, only the current working profile conversion object is displayed, because this conversion object is the latest.

Verify/Simulate

With the Verify and Simulate you can check the created migration object for programming errors and see how the data will get migrated running the migration object. During the Verify and Simulate a connection to the source environment is required. The connection details are stored per user and per profile. To view the connection details for the user click **Master Data → User.** If the connection details are not provided the DCA will use the connection details of the DCA admin user.

Verify All

Only the '**dcaadmin**' user can start verify all option. This option will verify all mapping objects for a profile which have the status Maintained, Error or Generated. The status of the objects is set to verified when successful else will get the status error.
During the conversion build also all objects are verified but then the status of the mapping object remains unchanged.

Conversion build

During a conversion build the mapping objects with the status "Expired", "Error", and "Comment Only" are skipped during the creation of the migration build.

Icons

For the most commonly used features from the menu bar in the mapping object screen an icon is available for quick access.

To create a new mapping object

This section describes a short procedure on how to create a mapping object. For a detailed description of the procedure see "Define conversion rules".

Open the maintain conversion mapping objects screen.

1. Select the mapping object for which the rules must be maintained. In case of creating a new mapping object the status of the mapping object is "**No-Rules**".
2. For a revision controlled profile the mapping object needs to be check-out first before it can be maintained.
3. Select **File - Open** for normal mapping objects, this will open the conversion object screen. For creating M-Script mapping objects first set the target source tables with the option **Define M-script**. When target tables are defined select the mapping object and use **File - Open**

4 In the conversion object screen set the source tables, add the precondition section and the main query section. Creating the content can be done manually or by using the build configurator. For legacy profiles the content must be written in VB programming language for Baan profiles the content must be created in the Baan4Gl language. Manually creating content can be done in the conversion object screen but also in a locally stored editor. To use the locally stored editor, click **Editor**.

5 Local mapping object functions can be created in the Function section screen. Click **Functions**.

6 Local variables/includes can be defined per mapping object. Click **Variables / Includes.**

7 After the main query section is defined the **Rules** tab is enabled. Enter the Rules tab. Select the dump number and create the mapping rules.

8 When the content is created click **Save** and **Exit** the mapping object screen.

9 Select the mapping object just maintained. Access **Specific-Verify** to verify the mapping object. Solve the programming errors until the status of the mapping object is verified.

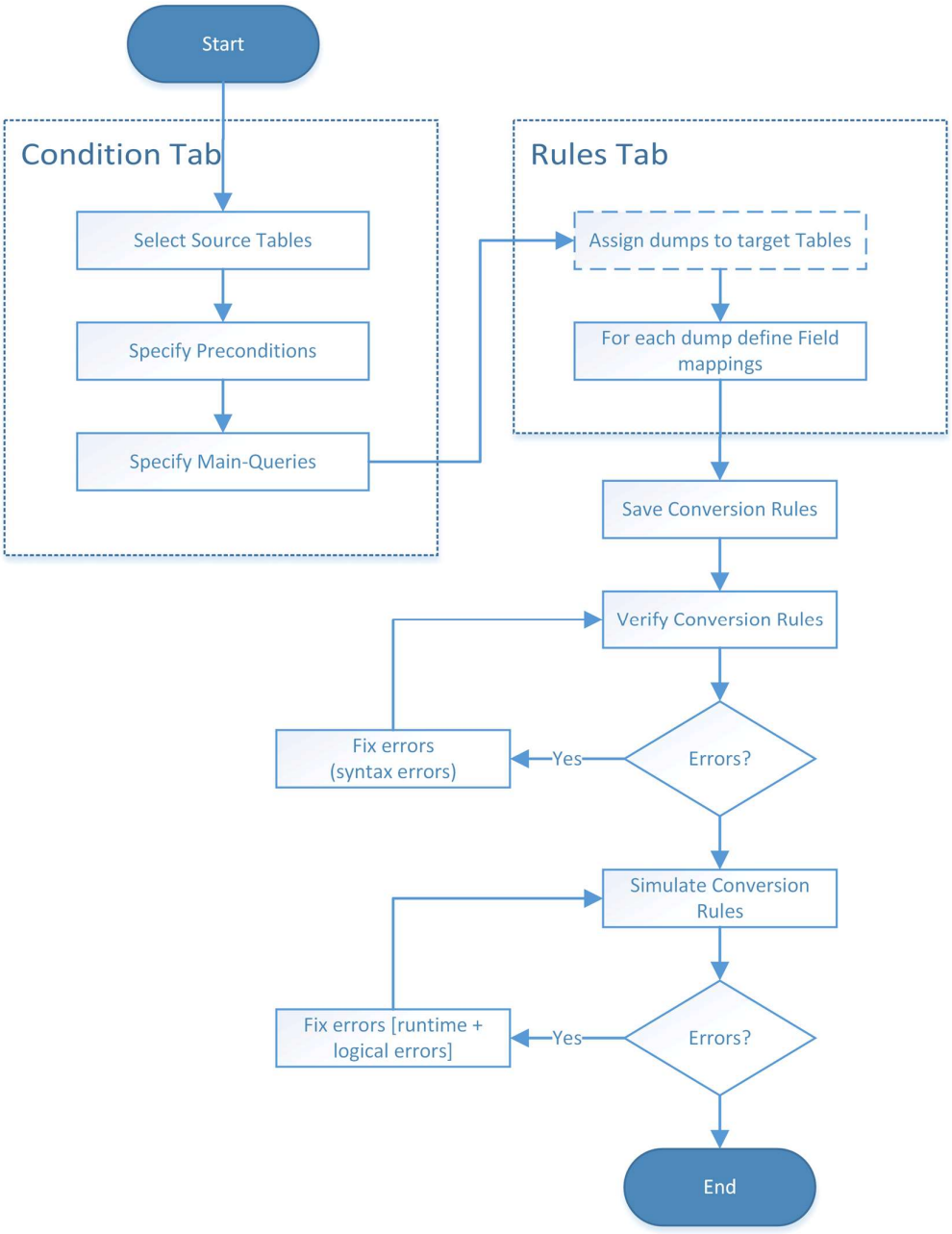
10 Select the mapping object just maintained. Access **Specific-Simulate** to simulate a data migration for the mapping object. During simulating the source data is used to create migrated data. The Converted data is shown in a new screen.

11 For **revision controlled** profiles check-in the mapping object.

Define conversion rules

The activities to define conversion rules for a single conversion object/M-Object are listed below. In the activity diagram, the boxes marked with dotted lines are only applicable to define the M-Object. While defining the rules, you must use VB scripting syntax. But for migration from one version of Baan to another, you must use the Baan Tools 3GL syntax.

A schematic flow of how to create or maintain conversion rules is shown in this diagram:

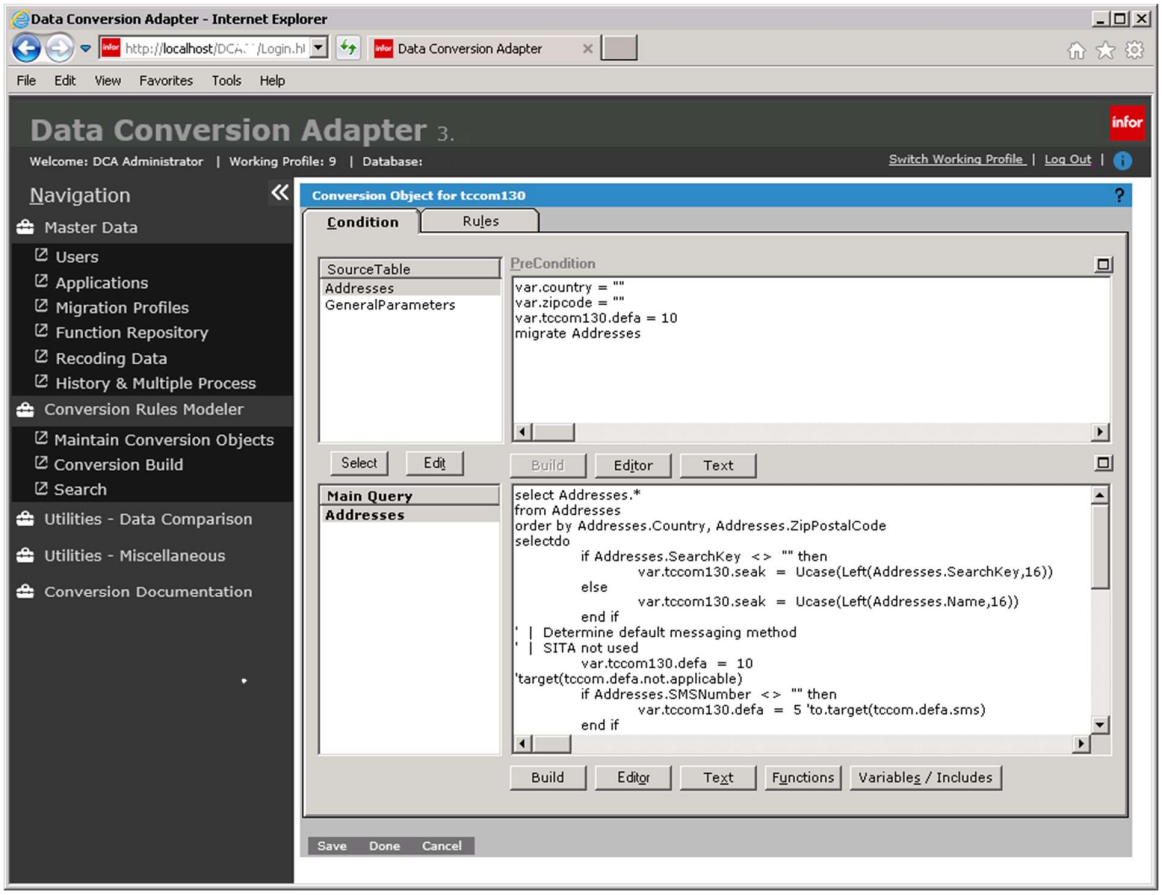


General functions within the mapping rules:

- Create and maintain Mapping rules.
- Copy current mapping rules to all other dumps within the mapping object.
- Copy the mapping rules from another dump in the same mapping object.
- Guess the mapping rule for all empty rules of a dump.
- Adding comments to the mapping rules used for creating migration Webhelp.

To define conversion objects:

1 Select a conversion object from the list and double-click to open the Conditional Editor pane, as shown in this screenshot.



The Conversion Rules Editor pane contains two tabs. The **Condition** tab enables you to perform activities following activities (see also the schematic rules process flow).

Select Source tables: You must select all the source tables to be used in the conversion rules. The source tables appear in the list box. To add new source tables, click **Select.** The source tables available in the source application are displayed. Select the required source tables and click **Done.** You can also manually add the source tables by clicking **Edit**.

Note: If the source tables that you require do not appear you might not have imported the source tables as a part to source application of your working profile.

2 **Specify the precondition:** Precondition is the high-level plan of migration logic where you can initialize variable, perform calculations for global variables, and select values in parameter tables of the source application that controls the migration logic. In this section, you can also use two key words of DCA, which have this meaning:

Migrate <*Source Table Name*>, which means you want to extract data from this table, and put into your target table. Using this key word will enable an entry in the Main Query section.

Dump <*Number*>, which means that at this point you want to write a record into the target table.

In this precondition section it is also possible to point to local defined functions within the mapping object or to global defined functions from the bcintdll0000 conversion script or function from the function repository in case of a Legacy profile.

You can write logic directly in the DCA or click **Editor** to start your local installed editor. You can define the path for the editor in the DCA Master data in the user screen.

3 **Specify Main Query:** For each **Migrate** tag in the precondition, you must maintain a query used to retrieve data from the source application. The queries in the DCA follow the ANSI SQL syntax.

The DCA introduces one section, namely: "**selectdo**…… **endselect**".

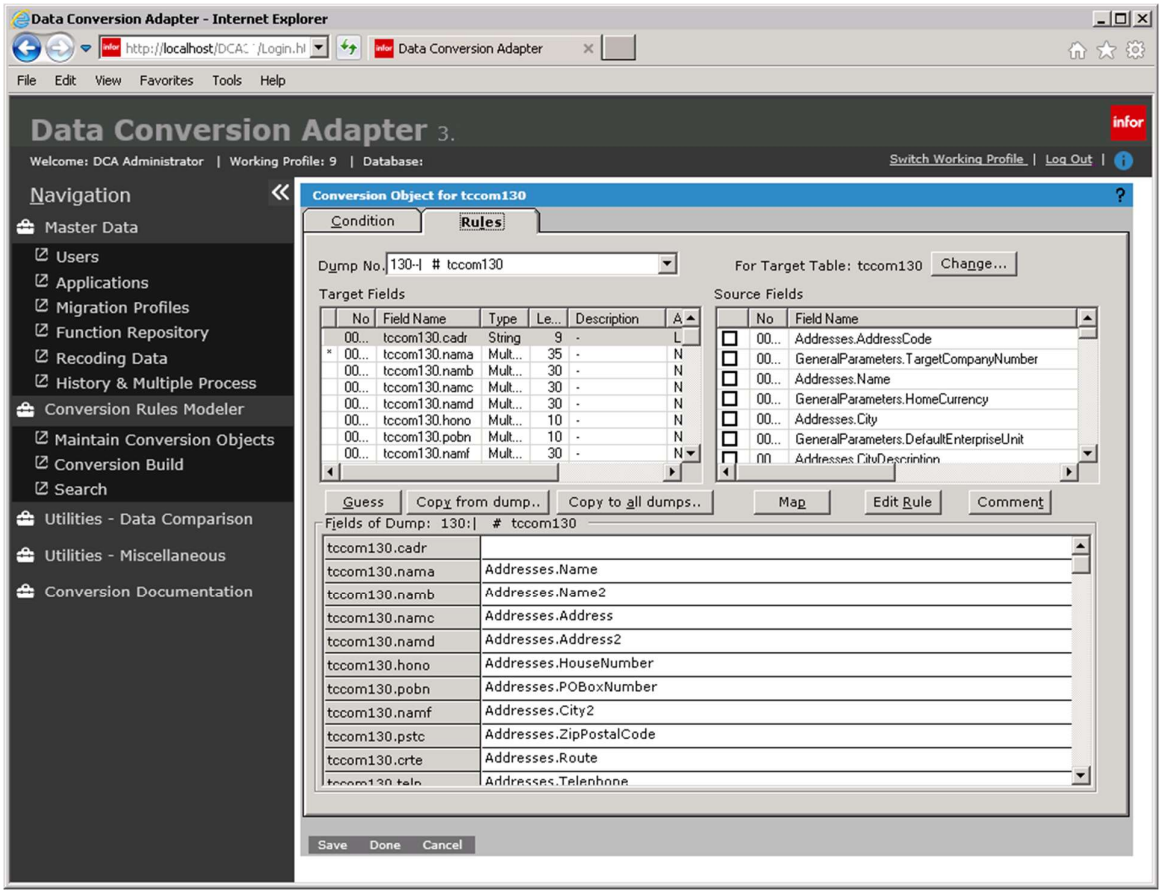
You can use this section to program what you want to do with each record you retrieve. In the Main Query, you can only use the keyword '**dump**'. You cannot use the keyword **migrate**.

In the main query section it is also possible to point to local defined functions within the mapping object or to global defined functions from the bcintdll0000 conversion script or function from the function repository in case of a Legacy profile.

You can write logic directly in the DCA or click **Editor** to start your local installed editor. You can define the path for the editor in the DCA Master data in the user screen.

You can also click **build** to use access the build logic section to automatically create a programming structure. For more information on the **build** functionality see the DCA help.

4 Click the **Rules** tab to open the Rules editor pane, as shown in this screenshot



The **Rules** tab enables you to perform activities 4 and 5 (from the schematic flow).

5 **Assign dumps to target tables:** This activity is required only if you define rules for an M-Object. In an M-object, you define conversion rules for a group of target tables together. Therefore, in your precondition or main query, you can have dump tags for each of these target tables. As a result, when you come to the **Rules** tab, to define mapping for each of the dump tags, as a prerequisite, you must associate the dump tag to a target table. This enables DCA to show, respective target table fields to be mapped when you choose a dump.

Note: While you edit an existing M-object, if the user changes the association of an existing dump to another target table, the previous map rules for that dump are lost while you save.

6 **Define field mappings for dumps**: You define field mappings for dump tags in the precondition and main queries. To define field mappings, select the dump from the **Dump No.** After you select the dump, the editor shows the corresponding mappings for that dump. The field mappings are maintained in the series of text boxes.

The various types of assignments available for a target field:

One or More Source Fields: You can map one more source table fields, with the proper joining operators.

Variable Assignment: You can assign a variable that is calculated in a precondition or a main query to a target table field.

Queries/Conditional Statements: You can provide queries as mapping to a target field.

Functions: You can use a function with a return value as a mapping rule. It is also possible to use some predefined mapping rule functions. For a detailed description of the available functions see the create logic pack work instructions. Some commonly used standard functions are:

- **Default()**: Use this function if the mapping rules needs to follow the initial value of the field. If no initial value is specified it will return an empty string or zero value depending on the field type.
- **Empty():** Use this function if you specifically want to use an empty value. Used to convert fields of type string.
- **Target():** Use this function specifically when an field of type enumerate needs to be converted and the source and target domain use the same enum names. However even if the enum names are the same the enum constant might be different. This function will handle the enum constant change.

In the case a new target RDD is loaded and the mapping object needs to be modified, it is possibly to use the guess function. This lets the DCA do a first guess of the mapping rule which then can be verified by the user.

For more details on the Guess, Copy from dump and copy to all dumps functionality see the DCA help. When developing a new migration logic pack also add a rules comment so it can be used to create logic pack help.

7 **Save conversion rules**: After you define the activities select source tables, define precondition, and Defined Main queries, you must save the conversion rules. Click **Save** to save the conversion rules. After you finish editing the conversion rules, click **Done** to save the conversion rules and exit the editor.

Note: If you edit conversion rules for a longer time, be sure to save the rules periodically. If you did not save the rules within the limit of your session time-out set in the IIS server, you lose the changes upon saving.

8 **Verify conversion rules:** For any legacy data migration, DCA generate VB scripts, which is an interpreted language. As a result, when you verify, DCA provides the first encountered error. To find out the syntax errors in the rules, you must perform a step-by-step process.

In case of BaanERP to Infor LN conversion rules, during verification, DCA tries to compile the script, and shows all the errors. In addition, in this scenario, to verify the conversion rules, you must have a FAT client installation of your source Baan release in the DCA server. For more information on FAT client installation, see the *DCA Installation Guide*.

To verify a conversion object, select the conversion object and, on the **Specific** menu, click **Verify**. Alternatively, you can also click **Verify** on the secondary tool bar.

9 **Simulate conversion rules**: You can use this option to identify runtime errors and logical error in the conversion rules. During simulation, DCA retrieves the data from the source database and formats the data according to the conversion rules and the results are displayed according to the target table format in a separate window.

To simulate a conversion object, select the conversion object and, on the **Specific** menu, click **Simulate**. Alternatively you can also click **Simulate** on the secondary tool bar.

Note: Simulate works on all the records in your source tables; so if the source tables' data volume is large; simulation takes longer duration to finish. In DCA master data/users, there is an option to limit the number of records in the results.

Special use cases

Case	Solution
You have a mapping object in a derived from profile but you must make a change.	You can only maintain mapping objects within your current working profile. Mark the required mapping object in the derived from profile. Access the **specific** menu option and choose **copy to current profile**. The selected mapping object is now also available for your current profile and you can start editing.
You made a change to a mapping object for a revision controlled profile but you want to undo the change.	Select the mapping object. Access **View** and choose **revisions**. A new screen will start with a list of all the revisions for this mapping object. Select the revision to which you want to revert back to and click **OK**. You will return to the mapping object screen, the object is check-out and the situation of the selected revision is restored.
You want to see the added comment to the mapping object.	Select the mapping object and access **View** and choose the option **comment**. This will allow you to view the comment. If you must maintain the comment access **Edit** and choose **comment**. You can also print the comment together with printing the content of the mapping object. Access **File** and choose **Print**. Mark the option "**with documentation**".

List of Special keywords/functions for DCA Studio Mapping Objects

Migrate *<Main Query Name>:*
This keyword triggers the DCA studio to create a Main Query. Within the main query you can define which source table(s) must be selected and define the mapping rules. This keyword can only be used in the main query section. You can define more than one Migrate statement. In general, the main query name is the main table in the select statement.

For example: Migrate BusinessPartner

Dump *<Number>*:
This Keyword will create a dump record for the mapping rules. Ones the dump statement is created the mapping rules can be defined for it on the rules tab. The dump number is often followed by a short description. This keyword can be used in the pre-condition section, in the main query section and also in the functions for these sections. This keyword should not be used in the mapping rules or functions for the mapping rules.
For example: dump 110 '|* Sold-to business partners

Default()
This function sets a default value for the mapping rule. During the generation of the script the initial value of the target field is taken.

When no initial value is available the studio sets a default value based upon the datatype:

- For datatype such as long, integers, and doubles the default is zero.
- For string data types the default in an empty string.
- For date datatypes the default is the zero date for a specified database.

The function default() can only be used in the mapping rule section.

Empty()
The function empty can only be used for string type data types. The output of the function is an empty string. The function empty() can only be used in the mapping rule section.

Target()
This function is used for enumerate datatypes. The output of the function is the constant value of the enumerate constant name. This function can also be used when the constant names between the target and the source environment are the same but the constant values are changed.
The function target() can be used in the mapping rule or a function which is used in the mapping rule.

For example: target(tcyesno.no)

Conversion Build

Use the Conversion Build process to generate the object pack. The object pack is generated based on your source application type:

- **Source application is Baan** → The conversion rules are translated into a Baan4GL program. The created program is compiled to create an executable for the Infor ERP source environment. The executable versions of the mapping object are used by the DCA dump

Engines which in turn migrates the data into flat data files. These data files can be loaded into Infor LN using the DCA loading engine.

- **Source application non-Baan** → The conversion rules are exported as XML files. These XML files are used by the DCA Legacy Engine for data migration. The conversion scripts are generated by the DCA Legacy Engine. The output of the DCA Legacy Engine is also flat data files which can be loaded into Infor LN using the DCA loading engine.

Only the '**dcaadmin**' user is authorized to run the conversion build. You can run the conversion build for one profile or multiple profiles simultaneously.

Note: During the conversion build process, you can also export the Logic Pack rules. This is helpful when you make weekly delivery of rules pack from your maintenance machine. The export option can be enabled in the build settings.

Click **Conversion Rules Modeler** → **Conversion Build**. The Conversion Build screen is displayed.

For a detailed description of the fields on the **Conversion Build** screen see the online help within the DCA34 studio.

When you selected for which profiles a Logic Pack Build is needed the user must specify the build settings per profile. The build setting page depends on the source application of the profile you have selected. For a detailed description of the build options see the online help in the DCA Studio in the build configuration screen.

For a conversion build of a Legacy profile these build options sections are available:

Profile, General, Build Range, Build Export

For a conversion build of a Baan profile these build options sections are available:

Profile, Database Driver, Script Optimization, General, Becs Detail, Server Information, Build Range, Build Export

In the Conversion build screen, you can click **Select** to select for which profiles a conversion build is required. Click **Settings** to define build settings per profile. To start the conversion build for the selected profiles, click **Build**.

Some of the main used build options are explained here. You can set these options per profile by clicking **Settings**.

The options at the **XML and Script Info** tab:

Profile (toggle):
With this option the user can create a build for only the mapping objects in a profile or create a full logic pack taking the profile derivation structure into account. With the latter option all the highest conversion objects are taken into the build.

In the General group these options can be set:

- **Consider checked out functions** (default = off).
  Specially used for revision controlled migration profiles during the development of a new Rules Pack. When this option is selected also checked-out mapping objects are considered during the build process.
- **Clear XML and Script Directory for Each Run** (default = on)
  Will remove previously created files for a previous build.
- **Consider History and Multiple process Setup** (default = off)
  Only required for a mapping object for which the History and Multiple Process is setup.

- **Include 2GB check** (default = on)
  This option is not applicable when creating builds for legacy profiles.
  Select this option when the source ERP Server is on Unix with a file size limitation. This option will add code to the generated mapping object to create flat data dump files with a maximum size of 2GB. For windows systems this is not required as there is no file limit size.
- **Is Multibyte Migration** (default = off)
  This option is not applicable when creating builds for a legacy profiles.
  Use this option when a target RDD is loaded from a customer coming from a Unicode environment. The changed mapping objects then needs to be compiled with this option set to yes.

  The options at the Compile “Info” tab. This tab is not applicable for legacy profiles.

- **Becs Information**
  Select the Infor BW configuration which must be used to compile the generated Migration object scripts. The selected BW configuration is a source environment on which the dump engine is installed.
- **Infor Server Information**
  Set some source environment server settings such as user credentials, package combination, and the system type. Specify also how the compile phase must be handled. These are the Compile modes:
  - **Normal**: Script is compiled in normal mode.
  - **Debug**: Script is compiled in debug mode. This mode is useful to debug a conversion script during the dump process at the source ERP side.
  - **Profiler**: Script is compiled in profile mode. This mode is useful to retrieve the performance statistics during execution of the script.

The options on the **Build Details** tab:

- **Build Range**
  Enter a selection range of the required mapping objects to consider in the build. You can select a specific object to be considered in the build.

  In the “Build Options” group you can set these options:

- **Generate XML**
  This option is not applicable for Legacy profiles.
  Select this option to create the xml versions of the conversion objects.
- **Generation Scripts**
  This option is not applicable for Legacy profiles.
  Use this option to generate the conversion scripts for the available (generated) xml versions of the mapping object.
- **Compile Scripts**
  This option is not applicable for Legacy profiles.
  use this option to compile the generated conversion scripts against the source environment.
- Export rules after build

  Determine if a Logic Pack Rules export is required. If this is selected the user must set the Up to Parent Profile.

When all the options mentioned earlier are selected they are handled in this order.
Generate the XML -> Generate Scripts from XML -> Compile Generated Scripts -> Export Logic Pack

To create a build

1. Open the Conversion build screen.
2. To select the profiles for which a build is required click **Select**.
3. Define the settings per profile by selecting the profile and click **Settings**. After the settings are completed the user can click **OK** on the Build Configuration screen.
4. If all profile settings are completed the user can start the build process by clicking **Build** on the build screen. A screen is displayed to monitor the progress.
5. After the build process is completed the user can view the output results and check the conversion build for errors.

Search

Using the Search option, you can generate various reports depending on your requirement.

The reports the user can create:

- Display all profiles in which the specific conversion object is present.
- Display all checked-out conversion objects in a given profile
- Display all conversion objects that are modified during a given period
- Display all conversion objects for which rules are not maintained
- Displays all conversion objects for which Field Mappings are not filled.

To search a migration profile:

Click **Conversion Rules Modeler → Search**. The Search screen is displayed,. Enter the required search criteria and Click **Search**. The result of the search is displayed in a new screen. For a detailed description of the fields on the **Search** screen see the online help within the DCA34 studio.

The search functionality is particularly handy for searching

- In which profiles does a specific mapping objects exists
- Get an overview of all mapping objects for a specific profile
- After an import of the latest Logic Pack rules search for all changed objects in the date range “Previous import date” – ‘current import date”.

Note:

When you select the option, *Empty Field Mappings*, the search operation may take long time depending upon the number of profiles you are searching and conversion objects in each of the selected profile.

Chapter 6 Utilities Data Comparison

The data comparison process enables a user to compare data models of two applications present in DCA Studio.

This module has additional options in case your source and target applications are of the type Baan.

- You can fine tune the data model comparison results and generate conversion rules for the changed and un-changed tables.
- You can also compare two migration profiles to indicate all the conversion objects in the source migration profiles that must be changed in accordance with the applications present in the target migration profile. This is useful when you have the standard migration profile, with Infor BaanERP BaanIVc4 Standard release to Infor LN Standard release. When you visit a customer site, you can create a profile with, say Infor ERP BaanIVc4 Customized to Infor LN Customized. Now you can compare the standard migration profile against the customization migration profile to identify all the conversion objects affected.

Compare DD's

The Compare DDs utility enables you to compare data models of two applications present in the Data Conversion Adapter. The comparison results are stored under a run number. It is not possible to compare an application of type Baan against non-Baan application.

To compare data dictionaries:

1. Click **Utilities – Data Comparison** → **Compare DDs**. The Compare DDs screen is displayed. For a detailed description of the fields on the **Compare DDs** screen see the online help within the DCA34 studio.
2. Fill in the source and target Application and choose the required Filter criteria.
3. Click **Compare** to start the comparison process. If the comparison is completed a result message will appear.
4. In case of a successful comparison run the output can be viewed. Click **View Results** or click cancel to leave this screen.

DDC Results

You can view the comparison results present in the system. When you compare two applications, the results are stored in Studio for later reference.

To access the DDC Results page, click **Utilities – Data Comparison → DDC Results**. The DDC Result screen appears. For a detailed description of the fields on the **DDC Results** screen see the online help within the DCA34 studio.

To view a comparison result stored in DCA:

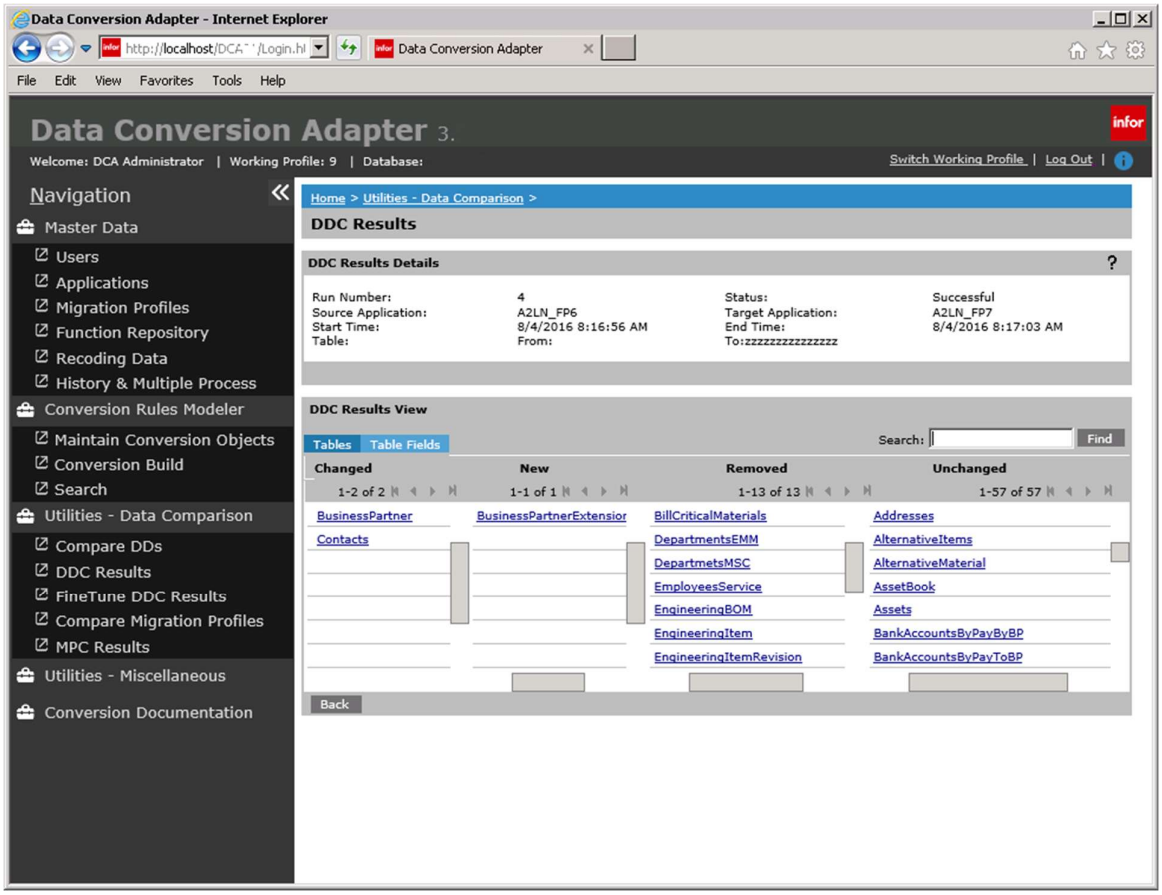
1. Click **Utilities – Data Comparison → DDC Results** to open the DDC results screen.
2. Select the Run number of the stored comparison result.
3. Click **Show Results**.

To print the output of a stored comparison result:

1. Click **Utilities – Data Comparison → DDC Results** to open the DDC results screen.
2. Select the Run number of the stored comparison result.
3. Click **Print Setup**. A new screen will appear to setup the print details.
4. Specify the selections required and click **OK**.
5. The results is displayed in a new screen. Using the **print this page** icon the output can be sent to a printer.

Show DDC Result

To view the DDC results, click **Show Results** to open the DDC Results screen. An example of a DDC result is shown in this picture:



The DD Results screen shows the comparison results for the selected run and the selected range of package, module, table, and domain. The details of the tables, table fields, and domains are grouped under the four categories: **Changed**, **New**, **Removed** and **Unchanged**.

To search for a specific table, table field, or domain, you can use the **Search** option. Enter the text you want to find and click **Find**. Click the tables, table fields, or domains to view the details.

There are two basic view options for the showing the results: By **table** or by **table fields**. Choose the view required by selecting the tab. In both tabs you can drill down the result by clicking the links in the **DDC Results View** pane.

Click a table field link to view the details of changes made in that table field. The Field Details pane is displayed with specific legend. The color red indicates a change in that attribute; green indicates that attribute is newly added; blue indicates removed attributes; and grey indicates attributes that are not changed.

By clicking on the domain hyper link the user can drill down to the changes made in the domain. This option is only applicable for baan applications.

Note: If you run the Compare DDs utility with the **Ignore Fields Number** as filtration criteria, the **Field Number** does not appear as **Changed** on the Field Details screen. Similarly, if you used the Compare DDs utility with **Ignore Combined Fields** as filtration criteria, the **Index** is not shown as **Changed** if the index is built using the same combined field at the source and target application side with various fields included as a part of combined field.

Fine-Tune DDC Result

This utility is applicable, only if the types of applications are Baan in a data model comparison run.

The Fine-Tune DDC Results utility enables you to fine tune the data dictionary comparison results stored under a specific run.

These examples illustrate how mismatches can yield incorrect results:

If a table X in the source application is renamed as table Y in the target application, the comparison result displays X as a Removed table and Y as a new table in the target application.

If a table X is fragmented into X1, X2, and X3, the comparison result displays table X as Removed, and X1, X2, and X3 as new tables in the target application.

Use fine tune to correct the above discrepancies in the comparison results. And you must fine tune only if want to generate conversion rules automatically for *Changed* and *Un-changed* tables.

To use the fine tune DD result follow the below steps:

1 Open the Fine-Tune DDC Results page: click **Utilities – Data Comparison → Fine Tune DDC Result**. The Fine Tune screen is displayed.

2 Click **Fine Tune** after you select the DDC run. The Fine Tune DDC Result screen is displayed. This page provides the details of the selected run. You must relate the removed source tables to new tables in the target application for the selected run. This page shows the removed tables and the new tables.

3 Create a link between the original table and the renamed table. Select the source table from the **Removed Tables** list and select the renamed table from the **New Tables** list.

4 Click **Create Link** to relate the two tables.

5 Click **Fine Tune** to alter the DDC result or click **Remove Link** to remove the link.

6 Repeat the process for all renamed and fragmented tables.

7 To compare data dictionaries, Data Model Analyzer tool can be used to compare two applications. These applications must be Baan applications.

DMA tool allows you to perform the following tasks:

- Compare RDD
- View Comparison Results
- Export Comparison Results

1 Starting DMA

2 Run the *DMAnalyzer.exe* to start working with the Data Model Analyzer tool.

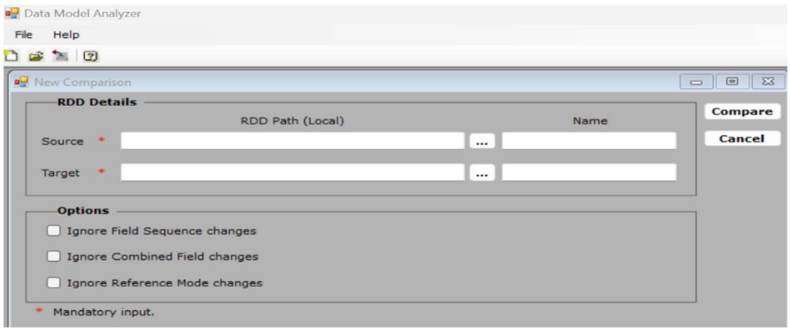
3 The **Data Model Analyzer** window appears:

4 You can use this screen to make a new comparison between source and target RDD.

5 To make a new comparison, perform the following steps:

6 Go to **File --> New** or click the Compare RDD icon . The **New Comparison** screen appears:

7 In the **RDD Details** section, specify the source and target RDD directory path. The name of the RDD must be logical for analyzing the comparison result. *For example, if you select RDD for Infor ERP LN source and target, you can name them as LN_Source and LN_target.* You can also select a source and target DD path by clicking the browse button.



This works similar to DDC comparison but its more user friendly and output is exported to excel which is more readable

Compare Migration Profiles

Use the utility to migrate data across Baan applications.

The Compare Migration Profile utility enables you to compare the similarity and differences between the two profiles (A2LN profiles are excluded from comparison). This functionality is useful for localization or customization of profiles that are created from the standard Rules pack.

If you create a migration profile that is derived from an existing profile with conversion rules maintained for conversion objects, similarity in the conversion objects of source and derived profile and this utility helps you to find the similarity and differences between two profiles. After you analyze the comparison result of migration profiles, you can copy all the changed conversion objects to the derived profile and modify the conversion rules, according to the differences.

To be able to compare profiles DDC results between source application of source profile and source application of target profile must be present. Similarly, DDC results between target application of source profile and target application of target profile must be present. If either is not found a message is displayed asking if it is ok to create the DDC results. If the comparison between profiles is already stored a question is asked to start the compare again or to see the already stored results.

To use the Compare Migration Profile utility, for a new compare:

1 Click **Utilities – Data Comparison → Compare Migration Profiles**. The Compare Migration Profile screen is displayed. For a detailed description of the fields on the **Compare Migration Profiles** screen see the online help within the DCA34 studio.

2 Select a source and target profile and click Compare.

3 A warning message is displayed if not all required DDC results are in place.

4 Click **Continue** to proceed with the comparison. A message is displayed after the comparison is completed.

5 Click **View** to see the result. The MPC Results screen is displayed.

6 Click **Show Results** to view the migration profile comparison results.

MPC Results

The Migration Profiles Comparison (MPC) Results utility maintains run numbers under which the results of migration profiles comparisons are stored. To view the results, you must select the run number. For more details see “Compare Migration Profile,” previously in this chapter.

The MPC Results session works the same as the DDC Results screens. By clicking the hyperlinks in the “Migration Profile Comparison Results View” screen the content of the mapping object is shown including the changed fields at the bottom.

Chapter 7 Utilities - Miscellaneous

This chapter describes the functionalities of the Utilities – Miscellaneous section in the DCA. The functionalities in this section of the DCA can be categorized into

- Where used functionality.
- Generate functionality.
- Uploading field description functionality.
- Import/Export functionality.

Where Used and Not used functionality

This utility enables you to search for table or table fields that are used or unused in the conversion objects of a specific migration profile.

To make a pattern search:

1 Click **Utilities – Miscellaneous → Where Used And Not Used**. The Pattern Searching screen is displayed. For a detailed description of the fields on the **Where Used And Not Used** screen see the online help within the DCA34 studio.

2 Enter the required Search criteria.

3 Click **Generate** to start the search process. On successful completion, the results are displayed on a separate screen.

Note: When you search all the table fields, in the complete range of conversion objects, the process continues for a long time. In this case you must search table fields pertaining to a specific package and module.

Generate Default Mappings

The Generate Default Mappings utility automatically generates conversion rules (conversion objects) for a migration profile. You must execute a data model comparison run between the source and target application of the selected migration profile, prior to the rules generation. Only the '**dcaadmin**' can perform the generation of conversion rules. The system ignores the target table for which conversion rules are already present. If this is the case the Guess functionality on the mapping Rules screen can be used to automatically complete the mapping rules.

The Generate functionality tries to find matching fields between target and source application. For evaluating matching fields for a mapping object it will take the source and target tables used for the specific mapping object into account.

For determining the mapping rule this order is used:

1. For each field in the target table, it searches the first source field with the exact matching table name and field name. If this search is not successful, then it searches the first field with the exact matching field name in any of the selected source tables. In case either of the search is successful, then the conversion rule is added as target table field=first source table field with the matching field name.
2. If the field with exact name is not found in the source table, then it searches for the field with the same domain in the source table. If found, the conversion rule is added as target table field=first source table field with the matching domain.
3. If the above rules cannot be met, the generate functionality will propose a default value by applying the “default()” function as the mapping rule. See the section “List of Special keywords/functions for DCA Studio Mapping Objects” for the flow of the “default()” function.

If the target field is of type UTC and the source field is of type Date or UTC then the conversion rules takes the UTC conversion into account.

After the mappings are generated the status of the generated mapping objects will show “generated” All generated mapping objects must be verified and completed.

To work with default mapping generation:

1. Click **Utilities – Miscellaneous → Generate Default Mappings** to open the Generate Default Mappings screen. For a detailed description of the fields on the **Generate Default Mappings** screen see the online help within the DCA34 studio.
2. Select the profile for which the mappings must be generated.
3. Define a required range.
4. Click **Generate** to generate the default mappings.

Note:

The generate functionality helps in creating conversion rules however the proposed rules still needs to be verified/approved and corrected where needed.

Upload Field Descriptions

When you create an application of type Baan in DCA Studio, the data models are imported from the Runtime Data Models. The runtime data models do not have the description of tables or fields. Use this utility to upload the table or field descriptions to a Baan application. Depending on the Baan release you must copy *getdesc.o* from the DCA server installation directory to the ERP server before you use this utility. For more installation details of *getdesc.o* see the Data Conversion Adapter 3.4 Studio – Installation Guide (U9094F US), chapter DCA34 Server Settings.

To load field’s description:

1. The object *getdesc.o* is installed at the ERP server side.

2 Click **Utilities – Miscellaneous → Upload Field Description**. The Load Field Description page is displayed. For a detailed description of the fields on the **Upload Field Description** screen see the online help within the DCA34 studio.

3 3Select the Infor ERP Configuration (BWC) file you want to use to connect to the ERP environment.

4 Select the baan application to which you want to load the table and field descriptions.

5 Click **Next**. The Connection Details screen is displayed

6Enter the password for specified user. If you want to connect as another user, change de BWC configuration file.

7 Specify the package combination to which the imported RDD (application) refers.

8 Click Finish to start importing the descriptions. A progress screen will appear. After loading the descriptions is completed a confirmation of a successful load is shown.

To view the loaded descriptions the user can use the browse functionality in the **Application** or view the description within the **Mapping object rules** screen.

Export Mapping Rules

The Export Mapping Rules utility enables you to export the conversion rules present under a migration profile to files together with source and target application. It is referred as the Rules pack for a migration scenario. You can import the Rules pack to another DCA server and customize it for a specific customer migration. To work with export mapping rules follow the below steps.

1 Go to **Utilities – Miscellaneous → Export Mapping Rules.** For a detailed description of the fields on the **Export Mapping Rules** screen see the online help within the DCA34 studio.

2 Select for which migration profile the export is required.

3 Enter the profile number up to which you want to export the conversion rules. To export the complete profile structure you must select the option ‘All parents’.

4 Choose if you want to export the revisions present for conversion object. Only applicable for revision controlled profiles.

5 Determine if you want to export all the latest conversion into one single profile. If this option is selected also a new profile number is required.

6 To start the export click **Export**. An Export complete message will appear on the screen when this process is finished.

7 The Exported files can be found in the folder <DCA Installation Directory>\DataConversionAdapter\data\ExportedRules\<exported profile number>. View the export log file called **ExportMR.log** within this directory.

Import Mapping Rules

The Import Mapping Rules is used to import the conversion rules from rules packs created by the Export Mapping Rules utility. Before using this utility you must unzip the Rules pack to a folder in the DCA Server. The unzipped Rules pack must contain the ExportMR.log file else importing the rules pack is not possible This file contains besides export details also parameters in how the export was created. This information is required by the import function. To work with import mapping rules, follow the below steps.

All the imported conversion objects will get as “Last modified user” the current user who is performing the import. The last modified date will remain the date of the latest change on the mapping object.

1. Go to **Utilities – Miscellaneous → Import Mapping Rules.** For a detailed description of the fields on the **Import Mapping Rules** screen see the online help within the DCA34 studio.
2. Browse to the folder where the rules pack are stored and click Open.

   The **Rules Pack Content** group box displays the profile as present in the Rules pack. The **Rules Pack Import Preferences** group box displays the order in which the profile is imported to the DCA Studio.
3. Click **Verify** to verify the import files.
4. If needed you can change the profile numbers, application names, and derived from profiles.
5. To start the import process click **Import**. A confirmation message is displayed after the process is completed.

Export Rules Patch

The Export Rules Patch utility enables you to export only the changes that occur during the maintenance of conversion rules. The changes can occur during a specific time frame, due to a specific issue or changes handled by a specific user. The exported changes are stored in a XML file. These changes (XML file) can be imported to any other DCA Studio, having similar profiles. This feature is available only for profiles that are revision controlled.

Create an “Export Rules Patch” by following the next steps.

1. Click **Utilities – Miscellaneous → Export Rules Patch.** The Export Rules Patch screen appears. For a detailed description of the fields on the **Export Rules Patch** screen see the online help within the DCA34 studio.
2. Select the **migration profile** for which you want to export the changes.
3. Select the **Ignore Exported** check box if you want to ignore the changes that are already exported, but satisfy the selection criteria you have specified.
4. Select the **Consider Includes** check box, in case you have modified functions (master data/function repository), as a part of fixing an issue. This option is only applicable for Legacy profiles.
5. Define a **Selection Range** by: Date, User ID and Issue ID or specify a specific set of objects required for the patch in the **Migration Object and Include** section.

6 To start the export process click Export. A confirmation message is displayed after the process is completed. Select **View** to view the export patch log file.

The exported patch files are stored in the folder <DCA Installation Directory>\DataConversionAdapter\data\Patch\<exported profile number>\

Import and apply patch

The Import and Apply Rules Patch utility enables you to import and apply the changes to a migration profile, from the XML file created by Export Rules patch utility. To start the import patch follow the below steps.

1 Click **Utilities – Miscellaneous** → **Import & Apply Rules Patch**. The Import & Apply Rules Patch screen is displayed. For a detailed description of the fields on the → **Import & Apply Rules Patch** screen see the online help within the DCA34 studio.

2 Select the migration profile to which you want to apply the changes.
Note: You must not maintain your own additional revisions in the profile to which you are applying the rules patch, because applying the rules patch will overwrite your revisions.

3 Enter the patch file name along with the path, which is present in the DCA server

4 Determine if you require a **Previous Revision** check.

5 Click **OK** to start the Import & Apply Rules Patch process. After the import is completed the import log file can be viewed.

Chapter 8 Conversion Documentation

You can maintain various conversion documents such as conceptual solution, technical documents, functional documents, and marketing materials for various conversion topics. You can also maintain various conversion topics with the help of the Conversion Topics page, and link conversion documents to various conversion topics with the Link Document page.

Conversion topics

You can maintain various conversion documents related to conceptual solution, technical or functional details, marketing, and so on for various conversion topics. Various conversion topics can be maintained with the help of Conversion Topics page and conversion documents can be linked to various conversion topics with the help of Link Document page. Before you can link a document to the added conversion topic the document must be uploaded first.

To add a Conversion Topic:

1. Go to **Conversion Documentation → Conversion Topics.** Click **New** to add a new conversion topic. This screen is displayed. For screen details see the online help within DCA34 studio.
2. Enter a Topic ID number, a description and some keywords
3. Click **Save** to store the Conversion Topic.
4. Click **Link Documents** to link uploaded documents to the conversion topic.

After a conversion topic is created you cannot change the ID number. Click **Modify** to change the Description and the Keywords.

Upload Document

You can maintain various conversion documents related to the profile such as: a conceptual solution, technical or functional details, marketing information, and so on. You can upload various conversion documents types such as: Word, Excel, PDF, or a plain text file, with the help of Conversion Documents page.

To upload documents:

1. Click **Conversion Documentation → Upload Documents**. The Conversion Documents screen is displayed. For screen details see the online help within DCA34 studio.

2 Click **New** to add a new document ID.

3 Enter the name of the document, the document type, the kind of document, some Keywords and the location of the file.

4 Click **Save** to upload and save the document ID.

5 To view the uploaded document click **View** .

Export object documentation

This functionality is used to generate Rules Pack Help in Web help format. With the Rules Pack web help a migration customer can view the migration rules comment details without installing the DCA Studio.

The Export Documentation process exports the comments on Mapping Objects and the comments on the Mapping Rules of the mapping object to a XML file. The XML file is created per profile but you can take into account the profile derived-from structure.

The documentation team processes the XML file to a web help format. For more details on this process, contact the Infor Documentation team at: documentation@infor.com.

To start the object documentation export:

1 Click **Conversion Documentation** → **Export Object Documentation**. The Export Object Documentation page is displayed. For screen details see the online help within DCA34 studio.

2 Choose for which profile (Rules Pack) you want to create the XML file.

3 Make a selection if needed depending on the requirements of the documentation team. Under normal circumstances you must run the session in full range mode.

4 Determine if you want to consider the comments of parent profiles as well.

5 Click **Export** to start the process.

6 After the process is finished a screen is displayed informing the user the XML files are created and shows the location. You can view an export log file by clicking **View Log File**.

Chapter 9 Migration Process Flow

This chapter represents the process flow for the activities that are performed in the migration process. All the activities and their sub activities are described in the process flow diagrams.

The process flow diagrams illustrate the steps that you must perform, such as define a user, define a migration profile, authorization users, define conversion rules, perform the conversion build, and so on. For more details, see each individual activity in this chapter. The process flow explains the activities within the scope of the DCA Studio. After the studio generates the object pack, the conversion is completed using the respective Engines. See chapter 2 for information about the Engines and the overall flow of a migration process.

Depending on the use of de DCA Studio you can define different flows. This chapter follows the process for creating a customized Object Pack for a migration customer.

The general process:

- **Install**
  - Install DCA Studio and check for updates
  - Create the DCA admin user
- **Import**
  - Import Logic Pack Rules (base)
  - Import Logic Pack Rules updates
- **Applications**
  - Define and load (customized) source application
  - Define and load (customized) target application
- **Profiles**
  - Define customized migration profile
  - Assign users to the profile
- **Mappings**
  - Define conversion rules
  - Verify
- **Build**
  - Create a customized Object Pack
  - Create a customized Rules Pack

Install

Before you can start with the DCA Studio you must install the product. For details on the installation see the Data Conversion Adapter 3.4 Studio – Installation Guide (U9094F US).

In this section a short overview is described.

Install DCA Studio and check for updates

Install the DCA34 Studio as described in the installation guide. After the installation of the DCA34 studio, check for updates on the Infor Service-Now Portal under KB**1126858.**

Install the update version.

After the DCA34 Studio is installed create the DCA34 super user '**dcaadmin**' using the User Manager from the Data Conversion Adapter menu.

Import

Retrieve the required Logic Pack Rules base and update version from Infor Service-Now portal. Which Logic Pack is needed depends on the migration process for the customer. The standard available Logic packs are:

- Solution: KB1585770, Infor BaanERP BaanIVc4sp15 Infor LN
- Solution: KB1642048, Infor BaanERP B50c sp17 to Infor LN
- Solution: KB1006397, Legacy ERP to Infor LN

Import Logic Pack Rules (base)

1. Retrieve the required version of the Logic Pack Rules (base) and save it on the DCA server.
2. Unzip the Logic Pack Rules.
3. Start the DCA Studio as the '**dcaadmin**' user
4. Go to Utilities – Miscellaneous → Import Mapping Rules
5. Import the Logic Pack Rules (base). This base version uploads all the required migration profiles including the applications and mapping objects.

Import Logic Pack Rules (update)

Currently the updates are delivered incremental solutions per supported migration logic pack and profile. All updates for the required profiles must be retrieved. For details and the latest installation guidelines always check the Logic Pack overview solution listed earlier in this chapter.

1. Determine which profiles are applicable for you customer
2. Retrieve the required versions of the Logic Pack Rules (update) and save it on the DCA server.
3. Unzip the Logic Pack Rules.
4. Go to Utilities – Miscellaneous → Import Mapping Rules
5. Import the Logic Pack Rules updates starting with the profile update for the lowest Infor LN Feature pack and work your way up.

The updates install the latest changed mapping objects per supported profile number.

Applications

With installing the Logic pack Rules also the base applications are loaded. As the requirement is to create a customized Object Pack create the required customized applications. In this scenario we consider a customer who has ERP customizations at the source side and on the target side.

Setting up an application depends on the type of application. For any legacy application, the data models are read directly through the OLEDB/ODBC provider, but for Baan applications, the data models are read from the runtime data dictionary [RDD]. An RDD is a set of ASCII files that explains the database structure of Baan.

Define and load source application

1. Click **Master Data** → **Applications** and click **New** to open the application screen.
2. Select the type of application you need (baan or non baan application). Follow the procedure to create a new application as described in chapter 4 Applications.
3. In case of a non baan application and the Legacy access data base is changed also create an ODBC connection to access the access database. If the legacy database is not changed then no must create a new application for it. For details on creating an ODBC connection see Appendix A.
4. After specifying the Application, load the RDD (baan application) or import the tables (legacy application). See chapter 4 Applications for details.

Define and load Target application

Handle the same procedure for the target application. Creating an ODBC connection for the target application is not required as the target application is always Infor LN.

Profiles

When making customizations to the standard Logic Packs it is always advised to create customized profiles for it. If you make direct changes to the delivered profiles then in case an update logic pack rule must be installed the update overwrites the customization changes made.

We recommend that you create a customized profile for general changed and an additional one for handling the recoding changes. Also create customized profiles if you must make changes but you do not need new applications.

Define customized migration profile

1 Create a customized profile for the new added source and or target application.

2 Derive the profile from the customized profile. If you are creating a new logic pack without a reference to existing profiles you can skip this step.

For details how to define a profile, see ““**Migration profile”** in the Master data Chapter .

Assign users to the profile

Log on to DCA as ‘**dcaadmin**’ user, and go to the migration profiles page. From the migration profile page, click **Users**. The **Associate or Remove Users** dialog box is displayed, which shows the users present in the DCA Studio and which user have authorization to the profile. Select the users you want to authorize for this profile, and click **OK**. You can select each user or select/deselect all the available users.

After the super user authorizes a user for a profile, only the user can switch to the new assigned profile to start working.

Mappings

Define conversion rules

For a detailed description on defining conversion rules see chapter 3, Maintain Conversion Objects.

Set the current working profile to the new created customized profile. Select the mapping object you must maintain. Copy the mapping object to the current profile and open the mapping object. In this case consider a just a change in the RDD as there are some fields added in the target table and no special other logic changes are required.

Go to the mapping rules table. The new additional fields will show a blank mapping rule(s). Click **Guess**. The DCA Studio tries to fill the mapping for you. The generated mapping rules must be validated. Complete the mapping rules manually.

Verify

During the Verify phase you must check if the created/changed migration rules for programming errors. This can be done by the verify functionality within the DCA. After the programming errors are fixed you must check if the data converts as expected/designed by using the Simulate functionality. This will pick up the source data and use the created conversion object to convert the data. If these checks are performed the mapping object can still contain functional errors, these can be detected within the post migration process. For the post migration process check the Data Conversion guide for the used migration path.

Verify conversion rules

For any legacy data migration, the DCA creates VB scripting for the created conversion objects. During the verification process the VB scripts are compiled and results are show to the user. The user should analyze and fix, if present, the reported syntax errors.

In case of conversion rules for a Baan/LN to LN logic pack, during verification, DCA tries to compile the script against the source environment, and shows all the errors. In addition, in this scenario, to verify the conversion rules, you must have a FAT client installation of your source Baan release in the DCA server. For more information on FAT client installation, see the *DCA Installation Guide*.

To verify a conversion object, select the conversion object and, on the **Specific** menu, click **Verify**. Alternatively, you can also click **Verify** on the secondary tool bar.

Simulate conversion rules

You can use this option to identify runtime errors and logical error in the conversion rules. During simulation, DCA retrieves the data from the source database and formats the data according to the conversion rules and the results are displayed according to the target table format in a separate window.

To simulate a conversion object, select the conversion object and, on the **Specific** menu, click **Simulate**. Alternatively you can also click **Simulate** on the secondary tool bar.

Caution: Simulate works on all the records in your source tables; so if the source tables' data volume is large; simulation takes longer duration to finish. Currently in DCA, there is no option to limit the number of records in the results. Currently to limit the result set, you can add a condition to your main query; remember to remove this condition, after the results are fine.

Build

The conversion build generates conversion scripts from XML files. Only the '**dcaadmin**' user can perform the conversion build process.
in case of a Legacy logic pack the created XML files will directly be used for the Legacy Dump engine.
In case of a Baan/LN Logic pack the conversion build also creates Baan4GL conversion scripts. The scripts are moved to the source environment and are compiled into executable objects. These executables is used by the source dump engines.
For more details on conversion builds, see "Conversion build," in Chapter 5, "Conversion Rules Modeler."

Create a customized Object Pack

Open the **Conversion build** screen in the **Conversion Rules Modeler** menu. Select the new created customized profile and click settings. Specify the required settings, ranges for your object pack. For testing the conversion build process first specify in the ranges one mapping object. If the conversion build process and the connection to the source ERP environment is ok, then run the process for all the mapping objects (full range) for the profile.

Create a customized Rules Pack

Creating a customized rules pack can be triggered from the conversion build screens. In the **Settings** screen on the tab **Build details** section **Build options** you can specify how and if a rules pack is required.

A rules pack can also be created without having to create a conversion build by starting the **Export Mappings Rules** screen in the **Utilities – Miscellaneous** menu.

Chapter 10 DCA Server Utilities

Besides the main application of the Data Conversion Adapter suite, the DCA Studio, there are also a few standalone applications which can be used on the DCA server. This chapter explains the various server utilities and their functionality. The additional server side utilities are:

- **DCA Legacy Engine:** Legacy dump engine to run the legacy object pack.
- **DB Config Utility:** Maintaining multiple DCA databases and switch database connections.
- **Load Datadictionary:** Loading Data dictionaries into the DCA Studio.
- **Recoding Utility:** For applying recoding to mapping objects.
- **Rules Recovery:** Recovery of mapping rules from xml files.
- **Script Debugger:** Debugger for legacy mapping objects (VB sources).
- **User Manager:** Maintaining users for the DCA Studio.

DCA Legacy Engine

With the DCA34 Conversion Adapter Suite also the DCA Legacy Engine is installed. With the DCA Legacy Engine and using the created legacy object pack, Legacy ERP data can be converted to LN Data. There is a standalone installer for the DCA Legacy Engine. The DCA Studio users such as consultants often installed also the DCA Legacy Engine this is now part of the standard DCA Studio installation.

NOTE:

Dumping Object ttxt010 will always report that only 1 record have been dumped.

The reason is that this object has the type binary.

For details about the use of the DCA Legacy Engine see the *Data Conversion Adapter 3.4 Legacy Engine - Installation guide (U9096G US)*.

DB Config Utility

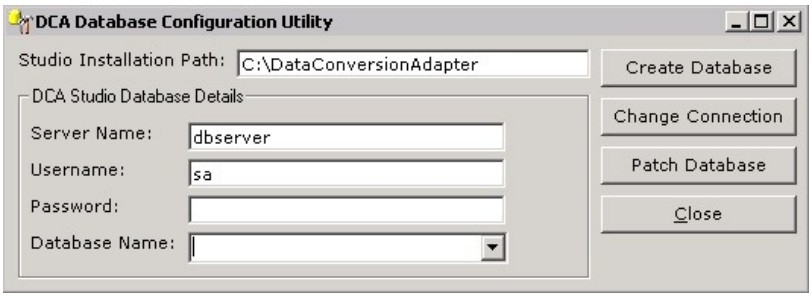
You can use this utility when you want to maintain multiple DCA databases. The DCA Studio can only be connected to one database at a time. With this utility you can create new DCA databases

and switch to other DCA databases and or Database servers. The DB Config Utility is also used in case the creation of the database during the installation process failed.

To start the DB Config Utility:

1 Logoff from the DCA Studio if this is running.

2 Run Start menu click **All Programs → Data Conversion Adapter → DB Config Utility.** As local Administrator. This will start the utility and the below screen is displayed. The installation directory, the database server name and the data base user is already defaulted.



3 Fill in the password of the database user and tab to the new field. A connection to the database server is made.

4 **Switch connection**: To switch to another existing DCA database, select the required database name and click **Change Connection**. During the switch the utility resets IIS which stops and starts the DCA connection to the database. After the change is completed the focus is returned to the utility and click **Close** to exit the program. When logging on to the DCA Studio a connection to the new database is established. The name of the connected database is shown in the top of the screen.

5 **Create new DB**: In case you want to create a new database for this database server enter a new database name and click **Create Database**. A confirmation message will appear telling the user the database is successfully created. After the database is created the user needs to switch to the new database. Click **Change Connection.**

Note: After you created a new database you must specify the DCA super user for the new database.

6 **Patch DB**: In case you want to link a data base of a previous DCA version. The previous database must be updated for new fields.

Load DataDictionary

This utility enables you to load Baan data models to a DCA's Baan application. Using this utility, you can selectively load pieces of the data models to the application. While you define a Baan application in DCA Studio, the entire data model for a specific release is loaded. The user does not have the option to select a part of the data model to be loaded.

This utility is specifically handy to load only the changed pieces of the data model in a Baan application. For example, one table or just the tables/domains from a specific package/module.

Note: For non-Baan applications, the selectively loading feature is present during importing of data models to application, and therefore the use of this utility is not required.

To start the **Load DataDictionary Utility**:

1. Run start **menu All Programs → Data Conversion Adapter → Load DataDictionary**. The Load Data Model Changes to DCA Studio window is displayed.
2. Fill in the Database server name, the database user and the password
3. Fill in the database name for which you want an additional load.
4. Go to the tab Data Model Details. A connection is made with the database. If this is successful the available application in this database are show in the application dialog box.
5. Select an application for which you want to upload RDD information.
6. You can directly load the RDD changes from the ERP Server or from a specified location on the DCA server. To select to load the RDD from the server the connection details to the server must be provided. If loading with the option RDD in local machine then only the path of the RDD files is required.
7. In the section Selection Range enter the range of packages, modules, and table codes which you want to upload in the selected application.
8. Select if you want to load only the tables or only the domains or both (tables and domains).
9. Click Update Data Model to start uploading the RDD for the selection into the DCA Studio application.
10. Use Load Data Dictionary when new Target RDD is uploaded for all tables. Sometimes, If the target RDD is uploaded via application load, the reference counters do not get uploaded correctly. Hence After loading via application, also load RDDs with this option.
11. Note: For text tables, do not select XML path.

If you use the same settings regularly then click **Save** to store the current settings. When restarting the utility with the Get defaults buttons the stored settings are retrieved.

Recoding

For the handling the recoding functionality the standard available migration profile needs to be adjusted. The recoding utility tool helps in automating the required modifications for the applicable migration mapping objects for allowing the recoding of source data.

During data migration from old systems, customers can also choose to standardize their master data codes, for example, address codes, warehouse codes, and so on.

The functionality can be used in the BaanIV to LN, B50 to LN, LN to LN migration paths and for the A2LN migration path.

The recoding utility searches the logic pack rules for target fields using a specific domain. The tool can replace the mapping rules for the found rules with a specific standard DCA function. This function will return a new value based upon an old value.
A list of old and new values is stored per recoding code in the source dump engine tables for the BaanERP/LN to LN migration paths and in the access template database for the A2LN migration path.

The power of this functionality is it recodes based upon domains and can apply the recoding to all mapping objects for a Rules Pack. For instance, the recoding of a warehouse is not only done

in the generic warehouse table but on all places where a warehouse code is used. Such as sales orders, purchase orders, warehouse orders, Production orders, invoicing, and so on.

The recoding utility can now also be used to automate the required changes when a unicode character correction must be applied for a specific LN target field. By using the predefined recoding code “UNIC” this utility will not apply the GetCode() function in the required mapping object but uses the bcint.dll0000.unicode.correction() function. As the Unicode character most of the time is only occurring for specific tables and fields the selection range on the form is extended. You can now specify for which field you require the change. At the dump engine side, you can define which character must be replaced with a new character.

The recoding task involves these activities in this order:

Step 1: Maintain conversion codes in DCA Studio. For more details on how to maintain conversion codes, see “Conversion table,” in Chapter 4, “Master data.”

Step 2: Apply conversion rules using the Recoding Utility. You can use this utility to apply the recoding function to field mappings in a specific profile. During applying the recoding a proposed recoding rules contains the standard DCA function GetCode(<Recoding code>, <Value to recode>). This function returns a new value based upon the recoding code and the current value. The user needs to setup the old and new values per recoding code in the DCA Dump Engine (Values by conversion code, table bcgct991 or bcdmp991) or in the access template database (ConversionCodeValues table).

This table shows an example of old and new values for the recoding code ABC:

Conversion Code	ABC
Old Value	New Value
A1	A0001
AB34	AB001
C1	AC001

During the actual migration dump the conversion rule GetCode(“ABC”, x) is executed and the new value for x is returned.

Therefore, during runtime, if the value of x is:

- A1, then A0001 is returned
- C1, then AC001 is returned
- B0001, then B0001 is returned. No value pair found.

If the function cannot find a specified old value then no recoding is applied and the function returns the old value.

When using a new customized profile to handle the recoding changes. The profile initially does not contain any mapping objects. During the Search step of the recoding the application linked to the profile is searched for target fields where the specified domain is used. For the found target field the mapping rules of the derived from profile is used to determine the new recoding conversion rule. When applying the recode for this field first the conversion object from the derived from profile us copied to the current profile. Then the new recoding mapping rule is applied to the mapping object of the selected recoding profile.

For more information about the screens in this task, see the “Details” section.

Step3: Maintain/generate recoding values the location depends on the Migration Path.

- For DCA source dump engines.
  For each of the conversion codes, you must maintain the old value and new values in the Values by Conversion Codes (bcgct9191m000/bcdmp9191m000) session. You can start this session from the Conversion Codes (bcgct9190m000/bcdmp9191m000) session. This session also includes features to generate new values, using a user-defined function, and former values from the table field.
- For A2LN migration path:
  For each of the conversion codes, you must maintain the old value and new values in the ConversionCodeValues table in the access database template of the logic pack. The user needs to add all required conversion code values.

Start the DCA Studio and go to the recoding data screen and setup the required recoding codes.

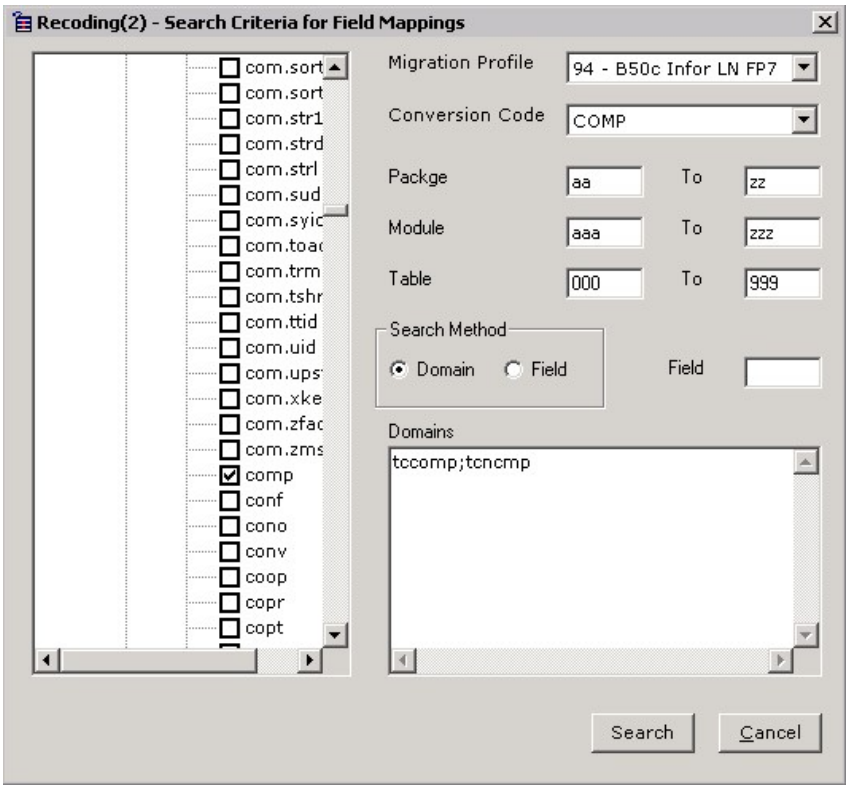
1 Create a customized profile derived from the standard profile to hold the recoding mapping rules changes.

2 Start the **Recoding Utility**:

3 Run start menu **All Programs → Data Conversion Adapter → Studio → Recoding Utility**

4 Check the defaulted values for Server Name, Database Name, and the database user

5 Enter the password for the database user and click **OK**. This launches the utility and the next screen is displayed.



6 Select the migration profile for which the recoding needs to be applied. During the recoding the mapping rules for this profile is modified.

7 Select the Recoding conversion code you want to apply.

8 Specify the range of the mapping objects to apply the recoding.

9 Specify or select the domains to consider for the recoding code. You can either select the domains by marking the checkbox or directly by typing the domains, without spaces, in the domain area. Using these domains, the complete target table fields in for the target application of the selected profile are searched. For example, the domains tccomp and tcncmp are used to define company numbers in Infor LN. Therefore, to recode/change company number during data conversion, you must enter these domains to identify target table fields, for which the recoding function must be applied.

10 Specify Search method by Field to consider for the recoding code. Filtered by a single Package- Module-Table Combination.

11 Click **Search** to start the recoding selection process. The results of the will appear on the screen. The results will show all the target fields using the specified domain and which fall in the specified mapping object range. The result will also show the actual rule and the new proposed recoding rule. The proposed recoding rules will contain the standard DCA function GetCode(<Recoding code>, <value to recode>). The function will return a new value based upon the recoding code and the current value. The user needs to setup the old and new values per recoding code in the DCA dump engine (table bcgct991/bcdmp991).

    If the conversion code “UNIC” is used then the proposed recoding rules will contain the standard DCA function bcint.dll0000.unicode.correction(”UNIC”, <value to recode>).

12 The user needs to verify if the proposed recoding is correct or not. If not then the user can deselect the **Apply Change** checkbox.

13 The user can **Export** the data to a (Tab) delimited File to edit the File with an external Text editor or import the (Tab) delimited File in Excel. The First (empty) Column has to remain and may not be removed. After doing the changes the user can decide to **Import** the changed data back into the recoding tool. This must be a (Tab) Delimited File also.

14 Click **Apply** to change the conversion rules for the target fields marked with **Apply Change** to **Yes**. The mapping rule is changed according to the proposed value in the selected recode profile. If the mapping object is not present in the recode profile then the mapping object is first copied from the derived from profile prior to applying the new recoded conversion rule.

15 To see the actual changed conversion rules start the DCA Studio and switch the working profile to the recoding profile. Select a changed mapping object and see the Rules.

Special attention

As the recoding uses the domains as a base in some case the recoding needs to be applied manually. Below are two cases where the recoding functionality fails.

Case1:

For instance, if a target field contains a generic string but depending on some fields the string can hold different entities (such as warehouses and work centers). In such a case the domain linked to the warehouse field will not detect the Generic String field containing a warehouse.

Case2:

Another exception is in case of using a variable as mapping rule where the mapping rules is a variable and the conversion consists of a conversion between a string and a long data type. When the source field is a long and the target field is a String. The used mapping variable is already converted from long to string. The proposed recoding rule will take only the variable into account. See also the below example.

Target field tirou003.tano of data type string and source field tirou003.tano of data type long.

Field rule using a variable: Field rule tirou003.tano: vartirou003.tano

The variable is converted from long to string in the main query section with adding a prefix. Such as vartirou003.tano = “A” + Str$(tirou003.tano)

An applied conversion recoding rules on field tano will result in: GetCode(“TANO”, vartirou003.tano). Here the variable is already converted into a string the GetCode lookup function cannot find the string as it expects a long value as input. The recode can be applied but the function GetCode in this case always returns the input variable.

This can be manually handled by applying this change in the main query section:

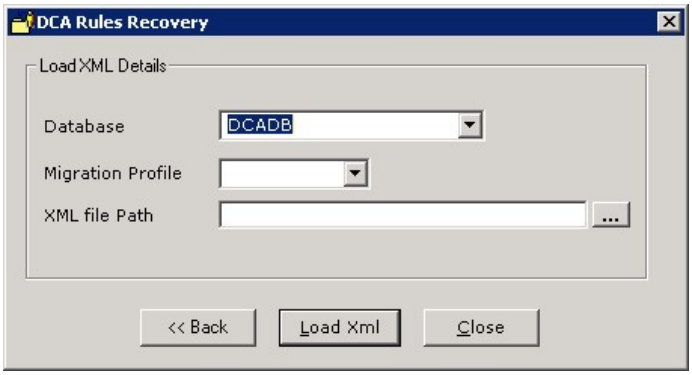
vartirou003.tano = “A” + Str$(GetCode(“TANO”,tirou003.tano))

Rules Recovery

In some rare occasions a mapping object can get corrupted. Use this utility to recover mapping rules from an xml in such case.

To start the **Rules Recovery Utility**:

1. Run start menu All Programs → Data Conversion Adapter → Rules Recovery. A startup screen is displayed
2. The Database server, database user is defaulted. Fill in the password for the database user and click **Next**. This screen is displayed.



3. Select the database and the migration profile which needs to be restored.
4. Specify the location of the (backup) xml file path to load.
5. Click **Load XML** and the mapping object is restored.
6. Click **Close** to close the application.

Script Debugger

This utility is useful when your data source is a Legacy ERP system. You use this utility to debug the VB script generated by DCA Studio during simulation. If the user simulates a conversion

object, the VB script file named, <*Conversion Object*>.vbs is created in the DCA Server's C:\Temp directory.

To use this utility you must enable the JITDEBUGGER in Microsoft Visual Studio Community 2022. Check if the script debugging is enabled. If on the DCA server also a full Visual Studio 2022 version is installed these steps can be skipped.

Download and install version 2019 or 2022(preferred).

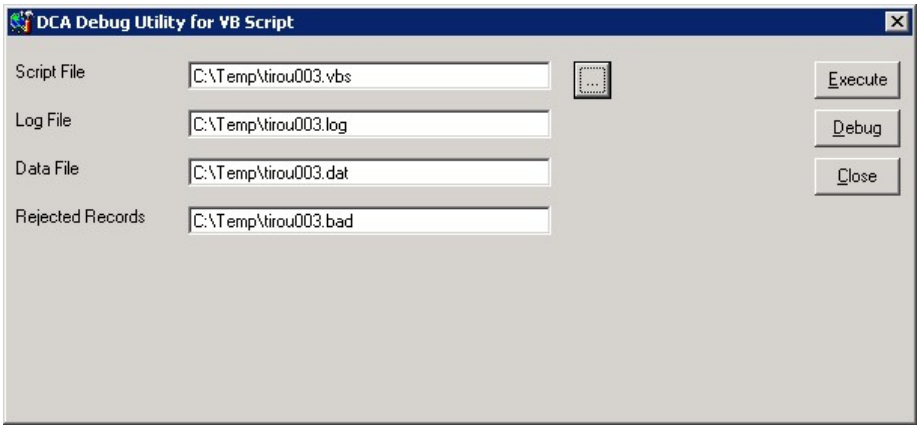
Microsoft Visual Studio Community 2022:

1 Download the Visual Studio Community 2022 with this link: https://visualstudio.microsoft.com/downloads

2 Install the Visual Studio Community 2022 on the DCA Server.

To start the **Script Debugger Utility**:

3 Run start **menu All Programs → Data Conversion Adapter → Script Debugger**. This screen is displayed.



4 Select the VBS script to debug from the C:\Temp directory with the browse **[…]** button.

5 Click **Execute** to run the selected conversion object's vbs file. Upon completion, the completion status/error messages appear in the file. You can view the data file for the retrieved data.

6 Click **Debug** to debug the VB script file. If you click this button, the VB Script file opens in the Microsoft Script Debugger, and you can go through the script, and see the values of variables.

   a In screen 'Choose Just-In-Time-Debugger'

   Select->New instance of Visual Studio Community 2022 , Press ok

   Using the following debug function keys:

F11 Step Into

F10 Step Over

F5 Continue

b For reloading the script without creating a New instance every time the following must be applied.

[VS2022] Menu->Debug->cscript Debug Properties

Update the fields with:

[Application]->Executable: C:\Windows\SysWOW64\cscript.exe

[Parameters]->Arguments: "//X C:\Temp\<*Conversion Object*>.vbs"

7 Using Ctrl+F5.

a In screen 'Choose Just-In-Time-Debugger'

Select->cscript - Microsoft Visual Studio Community 2022 , Press ok

You can repeat this step with Ctrl+F5 again, to restart over and over again.

8 Click **Close** to close the utility.

Microsoft Visual Studio Community 2019:

1 Download the Visual Studio Community 2019 with this link: https://visualstudio.microsoft.com/downloads

2 Install the Visual Studio Community 2019 on the DCA Server and ensure these additional settings are made:

a Desktop development with C++ [checked]

b Just-in-Time debugger [checked]

c WCF (Windows Communication Foundation) [checked]

(Installer versions may differ, WCF can also be found in ".NET desktop development".

In that case select and run 'Modify' for the second time

3 Start Visual Studio Community 2019 on the DCA Server. Go to Tools/Options from the main menu.

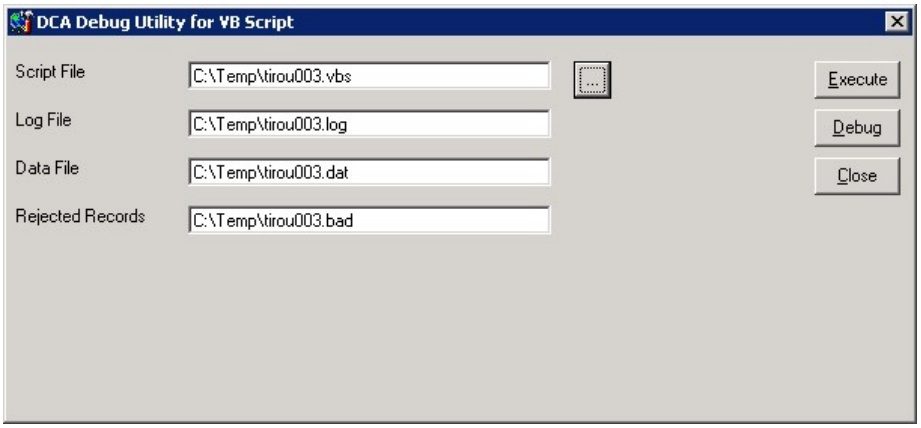
4 Go to Debugging / Just-in-Time.

5 The "enable just-in-time" debugger for script must be marked.

6 If not marked check this option or open a cmd shell and run vsjitdebugger.exe /regserver in the C:\Windows\Syswow64 directory.

To start the **Script Debugger Utility**:

7 Run start **menu All Programs → Data Conversion Adapter → Script Debugger**. This screen is displayed.



8 Select the VBS script to debug from the C:\Temp directory with the browse **[…]** button.

9 Click **Execute** to run the selected conversion object's vbs file. Upon completion, the completion status/error messages appear in the file. You can view the data file for the retrieved data.

10 Click **Debug** to debug the VB script file. If you click this button, the VB Script file opens in the Microsoft Script Debugger, and you can go through the script, and see the values of variables.

11 Click **Close** to close the utility.

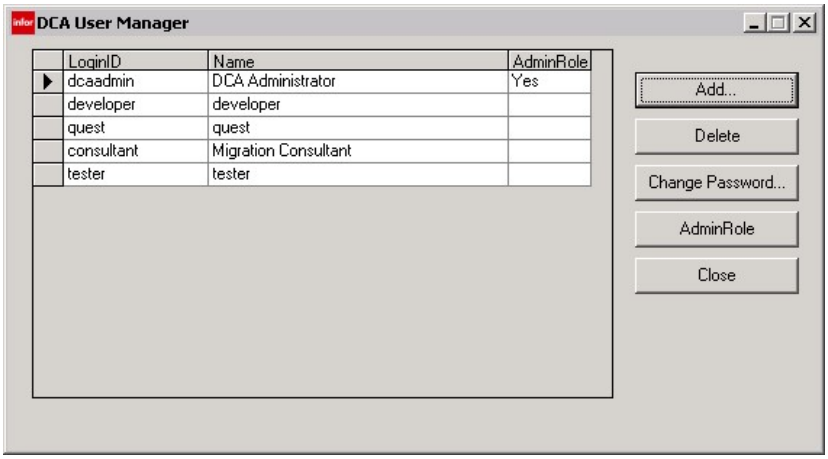
User Manager

You can use this utility to maintain users in DCA Studio.

To start the **User Manager Utility**:

1 Run start **menu All Programs → Data Conversion Adapter → User Manager**. The startup screen is displayed.

2 The Database server, the database user and the connected DCA database are display fields. Enter the password for the database user. This screen will appear.



3 Click **Add** to add users to the connected DCA database.

4 Click **Change Password…** to change the password for the selected user.

For more information about this utility, see “Create a DCA user.”

5 Click **AdminRole…** to change the role of the selected user.

Appendix A Creating an ODBC Connection

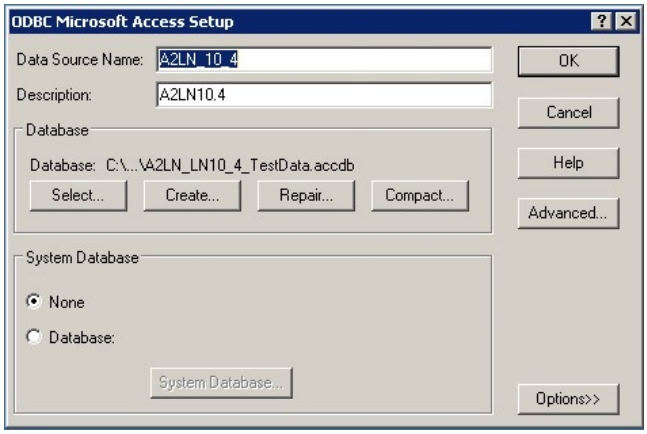
For legacy (non-baan) applications a connection to a source environment is required.

If you are using the standard delivered legacy logic pack, also called A2LN logic pack, the source data is stored in an Access database. A connection to this access database is required. It is also possible to define a logic pack to a Legacy ERP system for where the DCA directly makes a contact to the Legacy database. This appendix shows the activities to define the OBDC connection for both cases.

ODBC setup for Access Database

1. On the start menu Click **All programs → Data Conversion Adapter →Data Sources (ODBC) (32bit)**. The ODBC Data Source Administrator dialog box is displayed.
2. Go to tab **System DSN** and click **Add.**
3. Select **Microsoft Access Driver (*.mdb, *.accdb)** and click **Finish**
4. Enter a Data Source Name such as A2LN_10_<Version>. Enter a Description. Accept the default values. Click **Select** to link the access source data base to the created DSN.
5. Browse to the location of the provided Access database for the Legacy Logic Pack and select the database.
6. Click **OK** and the DSN is created
7. Link the DSN to the related Legacy Application in the DCA Studio.

With the DNS linked to the legacy application you can upload the data definition from access to the DCA Studio and to start simulations for the mapping objects.



Note: If during adding the ODBC connection only the option **Microsoft Access Driver (*.mdb,)** is available and not the option **Microsoft Access Driver (*.mdb, *.accdb)** then install this solution: Download ODBC Driver for SQL Server - ODBC Driver for SQL Server | Microsoft Learn

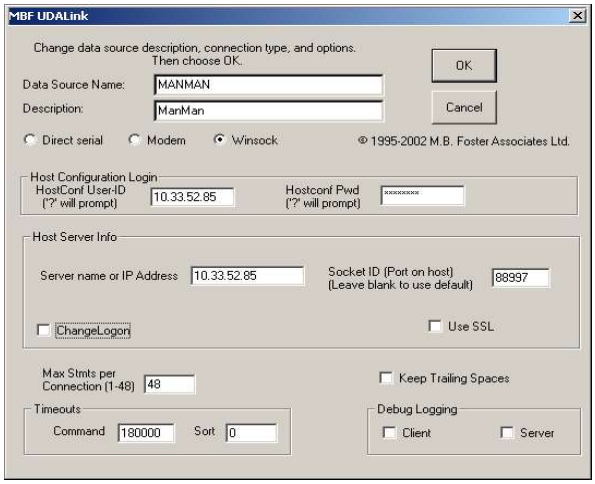
https://www.microsoft.com/en-us/download/details.aspx?id=13255

.
Use the x86 installer for 32-bit machines. After the installation repeat the procedure.
If the installation does not allow to install the 32 bit first try to uninstall the 64 bit version and try again.

ODBC setup for Legacy Database

To create a system Data Source Name (DSN) in the DCA Server.

1. On the start menu Click **All programs → Data Conversion Adapter →Data Sources (ODBC) (32bit)**. The ODBC Data Source Administrator dialog box is displayed.
2. Go to tab **System DSN** and click **Add.**
3. Select the required driver and click **Finish.**
4. A screen is displayed, according to the ODBC driver you selected.



5 Enter the details the driver uses to connect to the required database. The required details depend on the driver you use. For example, the MB Foster UDALink pane displays the information required to connect to an image database that runs on an HP server. The vendor MB Foster provides this driver.

6 Enter the details, such as data source name, host server information, and so on and click **OK** to create the DSN.

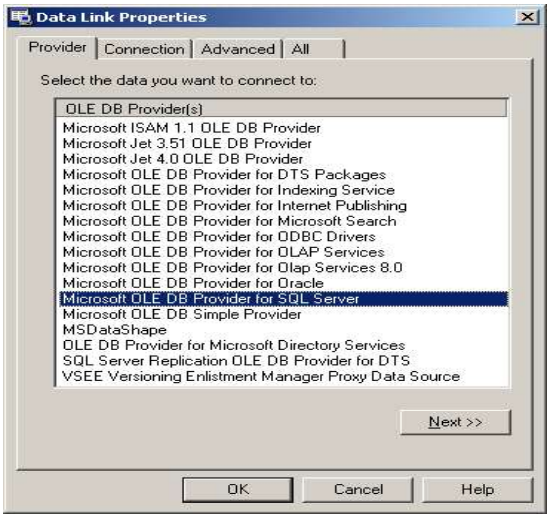
To install OLE DB

If you use an OLE DB driver, you must construct a connection string to connect to the database. To define the connection strings, you can use a predefined set of keywords. If you are not familiar with the connection string, see the procedure to construct the connection string:

1 Right-click to create a new text document on the desktop.

2 Rename the file extension from <*New Text Document*>.txt to <*New Text Document*>.udl.

3 Double-click the file to open the **Data Link Properties** dialog box, as shown in this screenshot:



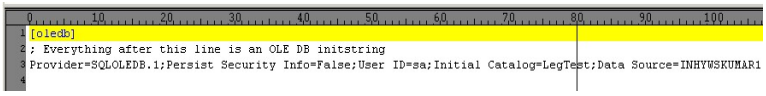
4 Select the OLE DB provider from the list and click **Next** to continue.

5 Enter the connection details. You can now test whether the connection is working properly.

6 Click **OK**. The system seeks confirmation to save the password in file.

7 Click **Yes**.

8 Open the file in a word editor, such as Notepad or UltraEdit. You can view the connection string as the third line:



Appendix B Lookup Database

Purpose

The basic objective of the lookup database is to help the customer maintain temporary tables for complex data migration process. These tables can be of types such as parameter, mapping tables, and so on. These types of tables are called lookup tables.

To define a lookup database

You must define the lookup database in the SQL Server database. Use the SQL enterprise manger to define the tables.

To access data in lookup database

To access the lookup database, you must create the required temporary tables in the lookup database. To obtain the data from the temporary tables, you can use these lookup functions:

Function Getvalue

Functions : GetValue	
Usage	GetValue(<Table name> as string, <Field name> as string, optional <Where condition> as string) as string
Input	Tablename: The temporary table name from which the value must be obtained. This is a mandatory field for the function. Fieldname: The field name for which the value must be retrieved from the temporary table. Where condition: The Where condition to be applied on the table. This field is an optional field.
Output	String value

Example

Consider a scenario for table bcbis700 with these records:

User	Udes	Type
Administrator	Administrator	admin
Bamarn	Amarnath	presales
cragha	Craghavesh	admin
kprasa	K.V.S.V. Prasad	cust

To get the value of the **udes** field for the user **kprasa,** you must pass these parameters to the function:

```
VarDesc = GetValue(“bcbis700”,”udes”,”unam=’kprasa’ ”)

GetValue returns the value “K.V.S.V.Prasad” to VarDesc

VarDesc = GetValue(“bcbis700”,”udes”)

GetValue returns the value “Administrator” to VarDesc
```

Note: If you do not provide the Where condition and more than one record is fetched, the first record is considered.

Example

If you want to use values from the outer queries for the GetValue function, use the values as follows:

```
select BAAN.SRCCUSTOMER.CUSNAME as var.cusname

from     BAAN.SRCCUSTOMER

selectdo

var.Bpid =  GetValue("tempCustomer","BPId","cusname = '" &  var.cusname & "'")

var.Addid =  GetValue("tempCustomer","AddId","cusname = '" &  var.cusname & "'")

     dump 1

endselect
```

Function GetValues

Functions : GetValues	
Usage	GetValues(<Tablename> as string, <Fieldname> as string, optional <Wherecondition>) as string, optional <Separator> as string) as string
Input	Tablename: The temporary name of the table from which the value must be obtained. This field is mandatory for the function. Fieldname: A field name or a set of field names for which the value must be retrieved from the temporary table. Where condition: The Where condition to be applied on the table. This field is an optional field. Separator: Enables you to separate the various fields provided to the function. This field is an optional field.

Output	String value

Example

Consider a scenario for a table bcbis700 that has these records:

User	Udes	Type
Administrator	Administrator	admin
bamarn	Amarnath	Presales
Cragha	Craghavesh	admin
Kprasa	K.V.S.V.Prasad	cust

To retrieve description and type, you can use this function:

```
VarValues = GetValues("bcbis700","desc~type","~","unam='kprasa'  ")
GetValues function returns the value "K.V.S.V.Prasad~cust" to VarValues
VarValues = GetValues("bcbis700","desc~type","~")
GetValues function returns the value "Administrator~admin" to VarValues
```

Note: If you do not provide the 'Where' condition and you fetch more than one record, the first record is considered.

Function GetSum

Functions : GetSum	
Usage	GetSum (<Tablename> as string, <Fieldname> as string, optional <Wherecondition> as string) as Double
Input	Tablename: The temporary table name from which the value must be obtained. This field is mandatory for the function. Fieldname: The field name for which the value must be retrieved from the temporary table. Wherecondition: The Where condition to be applied on the table. This field is an optional field.
Output	Numeric value

Example

Consider a scenario for the table bcbis702 with these records:

Rpid	Unam	Appl	Tapl
1	Kprasa	BaanIVc4	Baan60
2	Bamarn	verdi1	Baan61
3	Bamarn	BaanIVc4_old	Baan61
4	Bamarn	BaanIVc4	Baan61

If you want the sum of the **rpid** field, you must pass these parameters to the function:

```
VarSum = GetSum("bcbis702","rpid")
```

```
GetSum returns the value 10 to VarSum
```

Similarly, if you want the sum of the **rpid** field for the user **bamarn**, you must pass these parameters to the function:

```
VarSum = GetSum("bcbis702","rpid","unam='bamarn' ")
```

```
GetSum returns the value 9 to VarSum
```

Function GetAvg

Functions : GetAvg	
Usage	GetAvg(<Tablename> as string,<Fieldname> as string, optional <Wherecondition> as string) as Double
Input	Tablename: The temporary name of the table from which the value must be obtained. This field is mandatory for the function. Fieldname: The field name for which the value must be retrieved from the temporary table. Wherecondition: The Where condition to be applied on the table. This field is optional.
Output	Numeric value

Consider a scenario for the table bcbis702 that has these records:

Rpid	Unam	Appl	Tapl
1	Kprasa	BaanIVc4	baan60
2	Bamarn	verdi1	baan61
3	Bamarn	BaanIVc4_old	baan61
4	Bamarn	BaanIVc4	baan61

If you want the average of the **rpid** field, you must pass these parameters to the function:

```
VarAvg = GetAvg(“bcbis702”,”rpid”)
GetAvg returns the value 2.5 to VarAvg
```

Similarly, if you want the average of the **rpid** field for the user **bamarn**, you must pass these parameters to the function:

```
VarAvg = GetAvg (“bcbis702”,”rpid”,”unam=’bamarn’ “)
GetAvg returns the value 3 to VarAvg
```

Function GetMax

Functions : GetMax	
Usage	GetMax(<Tablename> as string, <Fieldname> as string, optional <Wherecondition> as string) as string
Input	Tablename: The temporary name of the table from which the value must be obtained. This field is mandatory for the function. Fieldname: The name of the field from which the value must be retrieved from the temporary table. Wherecondition: The Where condition to be applied on the table. This field is optional.
Output	String value

Example

Consider a scenario for the table bcbis702 with these records:

Rpid	Unam	Appl	Tapl
1	Kprasa	BaanIVc4	baan60
2	Bamarn	verdi1	baan61
3	Bamarn	BaanIVc4_old	baan61
4	Bamarn	BaanIVc4	baan61

If you want the maximum of the **rpid** field, you must pass these parameters to the function:

```
VarMax = GetMax("bcbis702","rpid")
```

```
GetMax returns the value 4 to VarMax
```

Similarly, if you want the maximum of the **rpid** field for the user **bamarn**, you must pass these parameters to the function:

```
VarMax = GetMax("bcbis702","rpid", "unam='bamarn' ")
```

```
GetMax returns the value 4 to VarMax
```

Function GetMin

Functions : GetMin	
Usage	GetMin (<Tablename> as string, <Fieldname> as string, optional <Wherecondition> as string) as string
Input	Tablename: The temporary name of the table from which the value must be obtained. This field is mandatory for the function. Fieldname: The name of the field for which the value must be retrieved from the temporary table. Wherecondition: The Where condition to be applied on the table. This field is optional.
Output	String value

Example

Consider the table bcbis702 with these records:

Rpid	Unam	Appl	Tapl
1	Kprasa	BaanIVc4	baan60
2	Bamarn	verdi1	baan61
3	Bamarn	BaanIVc4_old	baan61
4	Bamarn	BaanIVc4	baan61

If you want the minimum of the **rpid** field, you must pass these parameters to the function:

```
VarMin = GetMin(“bcbis702”,”rpid”)
```

```
GetMin function returns the value 1 to VarMin
```

Similarly, if you want the minimum of the **rpid** field for the user **bamarn**, these parameters must be passed to the function:

```
VarMin = GetMin(“bcbis702”,”rpid”,”unam=’bamarn’ “)
```

```
GetMin function returns the value 2 to VarMin.
```

Function GetCount

Functions : GetCount	
Usage	GetCount (<Tablename> as string, optional <Wherecondition> as string) as Long
Input	Tablename: The temporary name of the table from which the value must be obtained. This field is mandatory for the function. Wherecondition: The Where condition to be applied on the table. This field is optional.
Output	Numeric value

Example

Consider the scenario for the table bcbis702 with these records

Rpid	Unam	Appl	Tapl
1	Kprasa	BaanIVc4	baan60
2	Bamarn	verdi1	baan61
3	Bamarn	BaanIVc4_old	baan61
4	Bamarn	BaanIVc4	baan61

If you want a count of the records in bcbis702, you must pass these parameters to the function:

```
VarCount = GetCount(“bcbis702”)
```

```
GetCount function returns the value 4 to VarCount
```

Similarly, if you want the count of the records in bcbis702, for the user **bamarn**, you must pass these parameters to the function:

```
VarCount = GetCount (“bcbis702 , “unam=’bamarn’ ”)
```

GetCount function returns the value **3** to VarCount.

Appendix C Troubleshooting

This chapter describes the common issues and the troubleshooting technique to resolve these issues.

Errors using the DCA Studio

This table shows the DCA Studio error messages with specific cause, effect, and a solution.

Error Messages	Cause/ Effect/ Solution
Not able to select source table in the maintain mapping screen when connected to the DCA studio via a Windows7 System.	Cause: The DCA Studio installation is intended for a Window Server installation. Effect: An error message is displayed and the source tables cannot be selected from a Tree View. Solution: Manually add the source tables using the Editor button.
During the configuration build an error occurred during the compilation step: Message “Error in compilation: -<nr>-Method ‘~’ of object ‘~’ failed.	Cause: The DCA Studio is installed in a directory containing spaces. Effect: An error message is displayed and the user cannot compile or verify migration mapping objects. Solution: Reinstall the DCA Studio in a directory without spaces or remove spaces from the installation directory.
Search Empty rules is started but no output is generated and the process is running.	Cause: The Search is started using on a copied version of the DCA database. Effect: The search does not result in output. Solution: Export the complete profile including parents from your current Database. Create a new data base and import the exported rules. Start the search again.

Error Messages	Cause/ Effect/ Solution
The website cannot display this page. (Conversion Objects)	Cause: IIS setting is not correctly set. Effect: Cannot save large Mapping Objects. Solution: Change these settings with the IIS manager into: -Default Web Site->ASP->'Maximum Requesting Entity Body Limit' to 2147483647 -Default Web Site->ASP->'Response Buffering Limit' to 2147483647

Appendix D DCA 3.x Differences

This section gives a short over view of the enhanced features and fixes compared with the previous DCA version: DCA Conversion Adapter3.x Studio

If your previous version is DCA33 than only the enhancement/fixes for DCA34 are applicable

If your previous version is DCA31 (or earlier) than the enhancement/fixes for DCA34 and DCA33 and DCA32 are applicable.

Enhancements DCA3.2

General DCA Studio:

- **Enabled for MSSql2012:**
  The DCA3.2 version now supports MSSQL 2012 as a DCA Database.
- **Encrypted passwords:**
  The used passwords by the DCA Studio are now encrypted stored in global.asa file and shown with asterisks in DCA studio screens.

User Data:

- **DCA Admin User**:
  In the previous versions of the DCA it was always mandatory to create a default DCA admin user called ‘baan’. Only when the user was logged on as ‘baan’ certain functionalities where applicable, such as create a build or assigning users to a profile.
  With DCA 3.2 it is no longer required to use ‘baan’ as the DCA admin user. After you entered some users you can decide which user gets the DCA Admin role. For example, create user ‘dcaadmin’ and give this user the admin role. For more information, see chapter 4.
- **Syntax Highlighting**:
  When users are created the syntax highlighting is defaulted as “No” and if required can be switched on in the master data user screen in the DCA Studio. The required RICHTX32.ocx file is downloaded from the DCA server onto the client machine. It is not required anymore to separate install the file.
- **Show used ERP user for working profile**:
  When a user wants to verify a mapping object the compilation is done against the source environment. To make the connection some user data needs to be used. As this was not always maintained it was not always clear for the user which user data was taken. In this version the field ‘baan user’ will always be filled so the user can always see which user it taken for the compilation process. For more information, see chapter 4.

Applications:

- **Rollback:**
  When an error occurred during (re-)loading an application an unclear message was shown and the result was an empty application. In the current version we added a roll back mechanism. When an error, during the upload of RDD files into applications, occurred the original state of the application content is restored and a meaningful message is shown.
- **Load Complete:**
  When the upload of the RDD files into the application is successfully a message is shown telling the user the loading is completed.
- **Loading Descriptions for A2LN access table:**
  When loading an application for A2LN access then it might occur the description field of a field name was empty. This happed in case the description in access acceded the field length in the DCA studio application table. In this version when the above is applicable the description of the field will get the field name instead of being empty.

Conversion Objects:

- **Revision Date:**
  When a revision text is added after a change of a revision controlled mapping object now also the revision date is stored.
- **Remove migrate Statement:**
  When the “migrate …” Statement is removed the user will get a warning message if the applied changed needs to be handled. When a “migrate…” statement is removed also all the content related to it (Main query, dump statement, mapping rules) is removed. The additional warning message is to make sure removing the “Migrate” statement was on purpose.
  Remove dump statement:
  When the “Dump …” Statement is removed the user will get a warning message if the applied changed needs to be handled. When a “dump …” statement is removed also all the content related to it (mapping rules) is removed. The additional warning message is to make sure removing the “dump” statement was on purpose and not by accident.
- **Global Verify of Conversion Objects**:
  A global verify during the conversion build was possible however the status of the mapping object was not changed. In this version the ‘dcaadmin’ user can now verify all the mapping objects and based on the result, the statuses of the mapping objects are set accordingly.
- **Default view toggle:**
  The show mapping object screen as a default will now show the latest conversion object in the profile structure instead of all objects.

Script Generation:

String generation:
For the DCA3.2 version the generated code for converting strings is changed. Now default functions are used to convert strings and Multibyte strings. The conversion logic is moved to the source engine side. In this way the source engine can take into account the target database.
bcrun.convdll.string and bcrun.convdll.stringmb

- **Empty String:**
  The DCA3.2 now uses standard source function to handle the conversion of strings and multibyte strings. In case the source string is empty in previous version the value in the dump file would be a string with all spaces according to the field length.
  In case the source string is empty or the field can be mapped to an empty string then for MS SQL and Oracle databases the value in the dump file is one space, such as: “ “. For EDB as a database the empty sting will also be a complete empty string, such as: “”.

As a lot of these constructions are used the result is a reduction of dump file sizes and the record length.
bcrun.convdll.empty.string

Conversion Documentation:

- **Documents type changes:** When adding documents to the migration logic pack you can now also use the latest Microsoft word, excel format and PDF and a standard text file.
- **Authorization:** Conversion documentation can now also be added by all users and not only by the DCA admin user

Recoding Utility:

- The recoding functionality is now also available for the Legacy migration logic pack The same procedure can be followed as for a BaanIV/B50c/LN logic pack. The Recoding code values can be stored in the legacy access database.
- It is now possible to use the recoding utility for applying the conversion code “UNIC” when a Unicode character correction is required. Previously this had to be applied manually for each field and dump within a mapping object.
- During applying the prosed recoding change for the mapping objects the list can also be exported to Excel. In user can make modifications into the exported Excel file and import it again into the recoding utility tool. The changes are taken over and can be applied.

DB Config Utility:

- When a new DCA version is installed it is now possible to add/patch the used DCA database of the previous DCA version to the current version. A new patch button is added to the screen to apply the DB changes to the selected database.

About Information icon on main page:

- Shows the latest installed solution version number and copyright information.

Fixes DCA3.2

1. The “DB Config Utility” tool closed when wrong credentials for the db user were provided. Now the user will receive a message about the wrong credentials.
2. When the DCA database is located on a different machine it is not possible to load applications. This issue is prevented when installing the MSSQL2012 (64) NativeClient and Tools on the DCA Server machine.
3. It is now also possible to import an application with specific customer defined fields (CDF).
4. The DCA Studio installation does not require a reboot of the DCA server machine after installation. The message about restarting the server is removed.

5 When you click **Back** in the import rules session within the DCA Studio the session runs in an error. The error message is now collected and properly handled.

6 Several minor screen optimization issues resolved.
Wrong icon used, screen size, screen position, color top bar, removed redundant messages.

7 Due to some Microsoft security restrictions, sometimes it is not allowed to download cab files. The cab files are modified so they are not blocked for installation.

8 The build button on the Migration Mapping Object asp page was not functioning properly, this is restored and the build button functionality can be used again if required.

9 The install shield program of the DCA Studio assumed the build-in server “administrator” account was called “administrator”

10 Guess function improved for mapping objects of legacy profiles.

11 Mapping object status not always correct after check-in a verified mapping object.

12 Search functionality now excludes expired objects and check-out objects without changes. Also the performance of the search function is improved.

13 Within the mapping rules screen the fields were not always ordered as expected when the ordering was on field number. Now field number ordering is in the right order even if there are more fields than one hundred.

14 The performance of opening a mapping object is improved.

15 Resolved issues with array fields during the default script generation functionality.

16 Empty strings with full field length replaced by a string with one space for A2LN dump scripts and Baan4GL dump scripts.

Enhancements DCA3.3

General DCA Studio:

- **Enabled for MSSql2017 and MSExpress2017:**
  The DCA3.3 version now supports MSSQL 2017 and MSExpress2017 as a DCA Database.
- **User Manager can setup a user as DCA admin:**

  When a user needs to be setup for the DCA you can immediately mark the user as an DCA administrator.
- **Enhanced the upload capabilities of LN Data dictionary information:**

  The standard Data dictionary upload capabilities are extended with an XML file upload. This xml file can be generated using LN session “Export Table information for DCA” (ttmig1200m000). For more details see the DCA LN Engine user guide.

  Using this xml file the DCA retrieves import information on the Mapping Object/ table descriptions and field descriptions, the actual field sequence as defined in the LN Database and the applicable runtime tools fields. As the runtime tools fields are now available in the application it will also be possible to create mapping rules for these fields.
  This also made the “Row Versioning” setting in the DCA obsolete and is removed.
- **XI:**
  All web Pages of the DCA Studio have now a XI look and feel.
- **Guess Functionality**:

  The guess functionality to automatic generate mapping rules is simplified. Whenever the DCA can find an exact match between target field and source field than a 1:1 mapping is applied.
  In all other cases the default() is used as a mapping rule.

  The guess functionality is used during script generation Utility or via the Guess button on the Mapping object application screen on the Rules tab.

Master Data:

- **Import Baan Application:**

  Operating System: Windows
  After the import process additional Table Descriptions is added, if applicable in the xml import file. Therefor a XML Path (Enrich Database) can be populated. Using the xml file is optional. When using the XML file during uploading of a LN application it is no longer needed to separate run the “Upload field descriptions” utility.

Maintain Conversion Objects:

- **Conversion Object:**

  Table Field Descriptions are now visual in the Rules TAB from the Conversion Object

  Mapping Object/table description is now shown on the Mapping Objects overview screen and on the mapping detail screens.

  Initial Values column is added in the Rules tab as extra target and source application information.

  You can verify a mapping object in debug mode via the Specific menu option.

Load Data Dictionary Utility:

- Enhanced Target application upload.
  You can populate a XML Path (Enrich Database) field. Using the xml file is optional.

Security and Bug Fixes:

- Several Security and Bug Fixes are implemented.

Fixes DCA3.3

1. Recoding tool Failed if no Conversion Codes are available in the Database.

   Message:'Run-time error '9' Subscripted out of range'. When creating a new Profile from a previous export, revision texts is emptied now.

2. Mapping Object will not be damaged anymore after a Save if the Code contains

3. The user list (LoginID) in the 'Usermanager' is ordered and is now more convenient if there are a large number of users.

4. When doing an import in a no-baan situation the DCA notifies that the import was done successfully.
   Checking the tables we will see that they are still empty and the import has failed.

5. Function Repository will not open in the Editor.

   IExplorer->Security->Initialize and Script ActiveX Controls not marked as safe…
   Set option [Enable]

6. Import Tables for an Anything to LN situation.

   Sometimes if a large amount of tables must be imported, the length of the query can be exceed 2048 Characters.
   As result of this the submitted Query is rejected by the ISS server.

7. When using the recoding utility several issues are detected:

   - Applying recoding code "COMP" also applies it to fields such as tdsls304.cono (present in the baanIV logic pack)

   - The performance of applying the recoding code can be improved. It looks like the performance for DCA32 database which are converted from an earlier DCA version do not perform as it should.

   - The recoding is also applied to fields with default() and empty() mapping rule. These fields can be ignored for applying the recode

   - tffam module for some reason is not selected when applying recoding. Only when the object is copied to the recoding profile the tffam object is taken into account.

- When in the recoding utility only a search is done but not applied still all applicable objects are already copied to the recoding profile. In the recoding profile only objects should exist when the recoding is also applied within such an object.

- When the mapping rules consists of a multi-line if then else statement and recoding is applied to this field it leads to unpredictable results if the field variable declaration is not equal to the target field declaration.

8 Check-in Bug for ‘comment only’ objects.

9 Some aesthetic changes are made.

Enhancements DCA3.4

General DCA Studio:

- **Fill all empty Rules function on Profile level:**
  This new Function Searches all empty Rules and forces a value. The Function is applicable for Administrators only**.**

  The new function is located at: Conversion Rules Modeler->Search->Fill Empty Rules**:**
- **BaanUser Password length extended:**

  The BaanUser Password length may now have a maximum length of 25 Characters.
- **Treeview with Table info:**

  TreeView windows are resized and will populate Table Descriptions also.
- **Logged Users status window:**

  A new overview status window is added to view all logged users.

  This overview page is only applicable for Administrators.

  The new function is located at: Utilities - Miscellaneous > Users Logged In
- **DBOptions Admin check at startup:**

  To overcome latent User right problems an extra check has been added.
- **Trim trailing spaces reduced in .da2 and .dat Files:**

  Spaces are trimmed in .da2 and .dat to reduce File sizes and prevent maximum line lengths.
- **View Missing Domains per Profile:**

  Overview added to view all missing domains per Profile.

  The new function is located at: Master Data -> [Migration Profiles] Get Missing Domains (Administrators only)
- **Multi Language Support:**

  Multi Language (MLE) support in script generation

Security and Bug Fixes:

- Several Security and Bug Fixes are implemented.

Fixes DCA3.4

1 Read XML File, Caused Duplicates because of Combined Fields where skipped in the XML.

1 TrueView resize window/ Table Descriptions, will show all info now.

2 Fieldnames with uppercase characters will not be skipped in the mapping Objects anymore

3 XML Read, Only update/use of the first array-element of the Fieldname is done.

4 The import from already exported Profile with Refcntn Fields included in the Application are shown correctly under the Rules Tab in the Mapping Object window now.

5 Database patch for older Database available.

6 Hourglass, if DCA started from a client the hourglass is displayed in text instead of animated gif. Detected in: Verify object. Solution: Enable Activate Active-X settings in Internet/Custom Level

7 Change applications only applicable for admin users.

8 Error while printing mapping Object solved.

9 Change Guess functionality finetuned.

10 2 GB File size limit.

Improved tracking/detection of dat files who are exceeding 2GB size limit so files is

split more accurate if 2GB checkbox is checked.

11 Some aesthetic changes are made.

12 User with > 10 characters is not able to save changes in the DCA Studio.
13 A2LN Improve performance (MLE and CDF related).
14 A2LN MLE improvements.
15 Table "GeneralParameters" can not be imported from AccessDB.
16 Using the DCA Studio guess option, the created mapping rule for Date field types not correct.
17 A2LN Ref.counters not correctly created in migration script.
18 A2LN Improve performance, missing GetStr() in tttxt010 creation.
19 A2LN GetCode (ConversionCodesValue) will fail if target field is an empty string.
20 Password still visible in bcmap102 in mssql server.
21 Improve Recoding with Performance impact.
22 A2LN Recoding not working for multiple Table, must remain selected
23 Search/Filter Tables in A2LN Table import
24 In A2LN logic pack the function bcconvdll.unicode.correction() is called but function itself is not copied to the VBS script of conversion object
25 DCA Studio could not import DDs with CDFs which were MLE
26 DD Load utility Unable load Domain Information of a package
27 Latest build (DCA34_dca100023) introduced unpredictable results with the "Guess" option.