# Infor Enterprise Server Component Merge Tool User Guide

Release 10.7.x

Publication Information

Release: Infor LN 10.7.x
Publication Date: April 12, 2021
Document code: ln_10.7.x_lnttcompmergetoolug__en-us

# Contents

# About this guide

This document provides an overview of the Component Merge Tool (CMT) module's functionality and describes the functional procedures that are related to CMT.

## Intended audience

You need no detailed knowledge of the Infor LN software to read this document. However, you are more likely to understand the contents if you are familiar with:

- The overall structure of packages, modules, and sessions within the Infor LN Development Tools.
- The general business procedures used in everyday business practice.
- The basic concepts of enterprise resource planning.

For detailed descriptions of the module's sessions, see the Enterprise Engine online Help.

## Organization

This table shows the chapters of the guide:

| Section | Description |
|---|---|
| Chapter 1 | An Introduction to the User's Guide for Component Merge Tool (CMT) module in Infor LN Development Tools. It handles:<br>• The module's functionality<br>• The relationship of the module with other modules |
| Chapter 2 | The Component Merge Tool (CMT) procedure, if you want to know more about:<br>• The sessions in the procedure<br>• The results of the procedure<br>• The sessions that are related to the procedure<br>• The optional procedures that are related to the procedure |
| Chapter 3 | CMT using Update Sets, if you want to know more about:<br>• The sessions in the procedure |

# Contacting Infor

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

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

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

# Chapter 1: Introduction

The Infor LN or Baan software offers users the functionality to customize the Infor LN or Baan software to their own needs. Forms can be changed, menu's adapted, and new sessions can be developed.

However, if a new version of Infor LN or Baan is released, the customization of the old release must be transferred to the new standard. At this point the Component Merge Tool (CMT) can be used. In the following chapters is explained how to use the CMT module.

For detailed information of this module, comprehensive on-line Help is available on the system. Click **Help** and the on-line help information will be displayed.

Acronyms used in this document:

| | |
|---|---|
| ADV | Application Development |
| CMT | Component Merge Tool |
| VRC | Version Release Customization extension |

The Component Merge Tool is the CMT module in the Development Tools. This section provides information about:

- 
- CMT's functional procedures CMT as applied in the Development Tools
- The modules and business objects related to CMT

The Development Tools gives you the functionality to customize Infor LN or Baan to your own needs. This means that you can change or develop:

- Scripts
- Forms
- Menus
- Sessions

If a new version of Infor LN or Baan is released, you must transfer the customizations of the old release to the new standard. Note that CMT is not applicable with report scripts

# CMT as applied in the Development Tools

The function of the Component Merge Tool (CMT) module is to improve the process of merging customizations with a new Infor LN or Baan release. The CMT module supports merging of program scripts, libraries, and functions (“include” files).

CMT is part of the Application Development (ADV) module in the Development Tools. Some of the steps in the procedure contain sessions that are part of the Utilities in the Application Development (ADV) module.

# CMT’s functional procedures

The CMT module contains two procedures, which you can use to merge the changes in program scripts, libraries and function with a new version of the standard Infor LN or Baan components.

- Create target VRCs to transfer customized software components of an older Infor LN or Baan release to a new release.
- Using an Update Set to copy sources between Infor LN or Baan environments.

The first process will be discussed in Component Merge Tool on page 8, and the second will be discussed in CMT using update sets on page 15.

# Chapter 2: Component Merge Tool

This chapter describes how to use the CMT module.

This chapter describes:

- Merge procedure
- Prerequisites
- To carry out a merge
- To delete unchanged scripts in the source VRC

## Merge procedure

You can use the Component Merge Tool (CMT) procedure to merge customized software components of an older Infor LN or Baan release with a new release.

The procedure results in a new Infor LN or Baan environment in which the customized software components function properly.

## Prerequisites

### Master Data

Before you can use the CMT module, you must set up the master data. This involves the definition of a target VRC in the new Infor LN or Baan version, which includes the customizations. Customization of Infor LN or Baan is performed with package version release customization extensions (VRCs). A package VRC is in fact, a version of Infor LN or Baan.

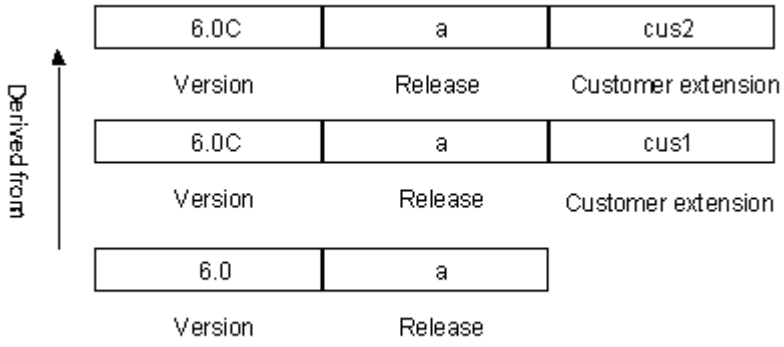
For more specific information about creating a new package VRC tree, see the *Administration guide* (U8854 US).

The package VRC tree structure is developed to avoid duplications by creating a parallel working environment. This procedure enables users to work in their own VRCs while developers can continue

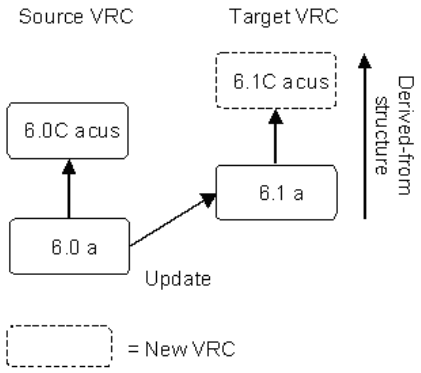
customizing the applications at the same time. This means that the user can be working in a package VRC that contains the standard software of a package while the developer is working in a package VRC that contains the customizations for that package.

A new package VRC is derived from a previous VRC and it only contains the software components that have changed compared to the previous VRCs. This is done to avoid duplication.

This figure shows how the package VRC is derived from a previous package VRC.




If a new release of Infor LN or Baan is introduced, you must copy the existing customizations such as, menus, forms, sessions, domains, and scripts to the new release. The CMT module is developed to simplify this process of merging customer-specific customizations with a new Infor LN or Baan release while preserving the customizations of the old release.




There is a distinction between regular updates and updates with a derived-from structure. The CMT module does not play any part in the regular update process. The CMT module is specifically developed for updates with a derived-from structure. For these updates, a merge takes place between the customized software components of the previous package VRCs and the new Infor LN or Baan release.

If the merge process is completed, the new version and the original version share the same derived-from structure, which enables the user to use the customizations in the new Infor LN or Baan release.

The following conditions must be met to successfully merge the customizations:

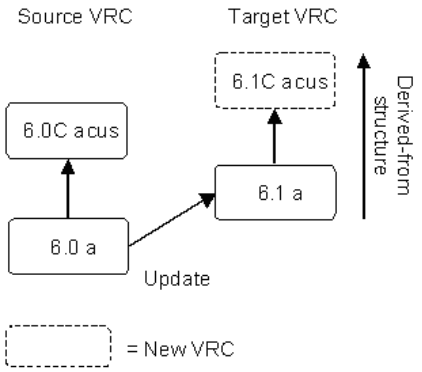
- The Infor LN or Baan standard release is already updated to the new standard release.

- The derived-from structure for the target VRC must exist in the new Infor LN or Baan release.

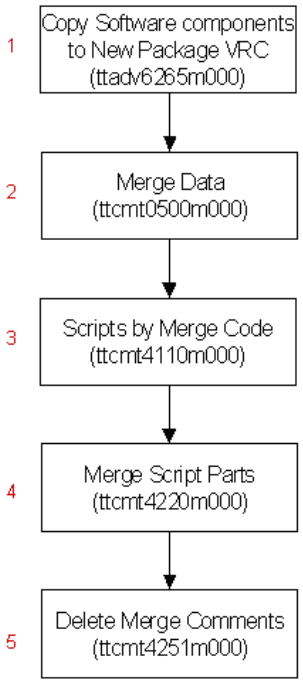
## To carry out a merge

Before starting the merge procedure of the customizations with a new Infor LN or Baan release, you must create a derived-from structure for the VRCs in the new release. The derived-from structure must comply with the derived-from structure in the previous release.

This diagram shows the derived-from structure of package VRCs.




This diagram shows the merge procedure:




The merge procedure consists of these steps:

1 **Copy Software Components to New Package VRC** (ttadv6265m000).

When the new derived-from structure is created, you must use this session to copy the customizations of the source VRC to the target VRC. This enables the CMT module to link the old customizations to the new functionality of the Infor LN or Baan release.

2 **Merge Data (ttcmt0500m000)**. You can use this session to enter the master data for the merge procedure to prepare for the actual merge.

The master data includes:

- A merge code that identifies the merge procedure
- The source VRC that contains the old customizations that will be released
- The new target VRC to which the customizations will be copied
- The package of the customized software components
- An ident flag
- An ident string

Although you can create a merge for up to ten packages, the customer extension of the source and target VRCs of the specified packages must be the same. You can enter an ident flag and string on the **Parameters** tab in the details session. The **ident flag** determines whether the ident string is placed in the components that are merged.

- If the **Ident flag** is set to **`All scripts`**, the ident string is added to the scripts that are selected for the merge process.
- If the **Ident flag** is set to **`Only Changed Scripts`**, the ident string is only added to those scripts, which are actually merged.

- If the **Ident flag** is set to **`No`**, no ident string is added to the scripts.

The ident string can contain one or a combination, of these variables:

- **$user**

  This variable is substituted by the user name in the Merge Script Parts (ttcmt4220m000) session.

- **$date**

  This variable is substituted by the current date during the merge process.

- **$vrc**

  This variable is substituted by the target VRC.

The syntax of an ident string that is used in the merge process must adhere to certain syntax. For example, an ident string that includes the $vrc and $user variables:

# ident “@Update: $vrc,$user”

**3 Scripts by Merge Code** (ttcmt4110m000)

You can use this session to define the components that must be merged by linking them to connect the scripts in the source VRC to the merge code.

In this session you can define:

- The merge code that is defined in the **Maintain Merge Data** (ttcmt0500m000) session
- The type of script. For example, a function, library, or program script

The **Script Type** field indicates the type of script that will be merged.

- If the **Script Type** is **`Function`**, the script that will be merged is a function and the existence of the script is checked in the Functions table.
- If the **Script Type** is **`Library`**, the script that will be merged is a library (DLL), and the existence of the script is checked in the Program Scripts table.
- If the **Script Type** is **`Program Script`**, the script that will be merged is a program script, and the existence of the script is checked in the Program Scripts table.

The description of the script is copied from the Program Scripts table or Functions table, depending on the script type.

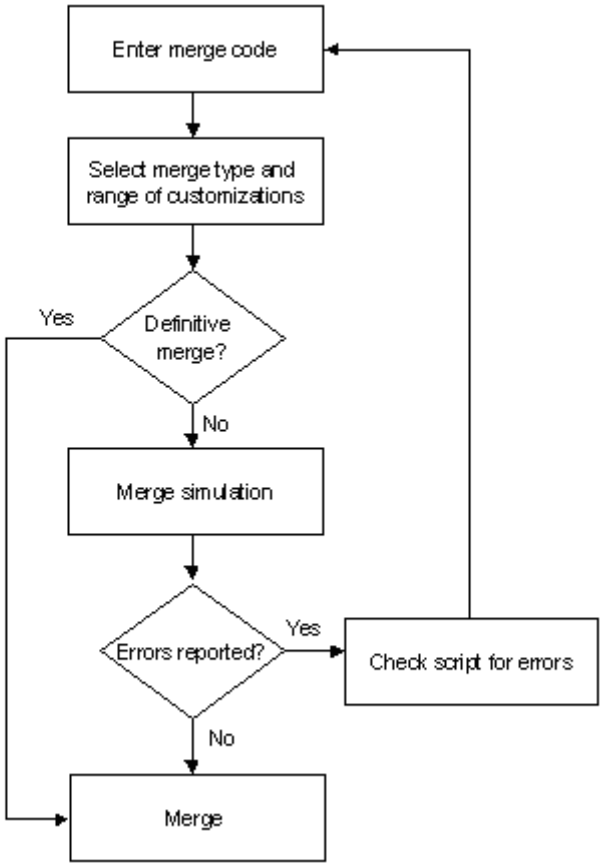
The Status field indicates the status of a script during the merge process. The status is entered in this session, after which the status is changed by using the **Merge Script Parts** (ttcmt4220m000) session.

You can only merge scripts that are connected to a merge code.

**4 Merge Script Parts** (ttcmt4220m000)

You can use this session to actually merge the scripts. You can select functions, libraries, and scripts to be included in the merge process. You can also preview the results of the merge without changing the source or print the results after merging.

This diagram shows the definitive merge procedure.




You can simulate the merge procedure by clearing the **Definitive Merging** check box. The merge script will be simulated and the result of the merge process is displayed as a report. If no errors are reported, you can select the Definitive Merging check box and the Reprocess check box to perform the actual merger.

If the **Definitive Merging** check box is selected, the selected function, library, or program script is stored in a temporary file. If that same function, library, or program script is encountered in the target VRC, it is replaced by the contents of this temporary file.

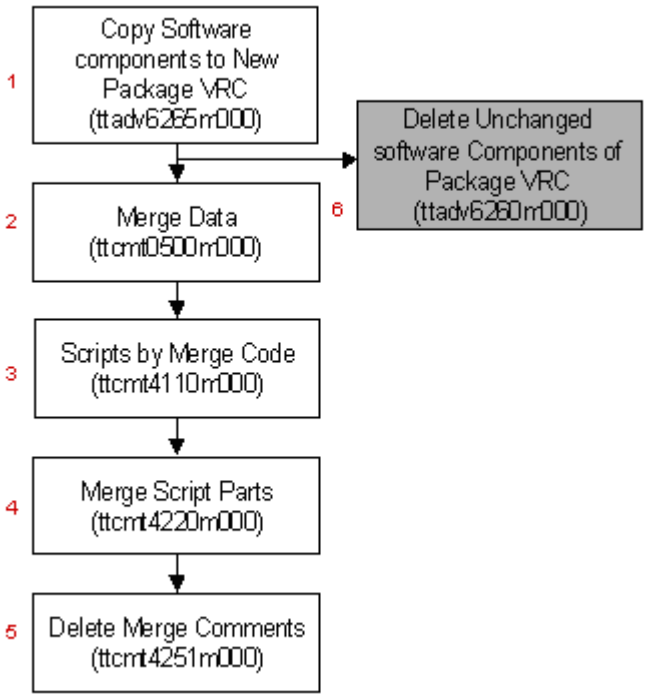
If the **Reprocess** check box is selected, scripts with the *Merged* status or *Ident* status are reprocessed. Scripts with *Ident* status or *Not Reprocessed* status, can result in a duplicate ident string.

**5 Delete Merge Comments** (ttcmt4251m000)

During the merging procedure, the **Merge Script Parts** (ttcmt4220m000) session adds comments to the merged scripts in the target VRC. Remove these comments to avoid problems during a future merger. If the merged scripts function properly in the target VRC, you can use this session to delete these comments.

# Delete unchanged scripts in the source VRC

The delete unchanged software components of package VRC procedure consists of one single step:




### Delete Unchanged Software Components of Package VRC (ttadv6260m000)

Copy the software components that must be merged to the target VRC. You can use this session to delete unchanged scripts in the source VRC.

# Chapter 3: CMT using update sets

The previous chapter describes the normal use of CMT if a full customization VRC must be upgraded to a new version of the standard software. All VRCs and software components involved had to be located in the same BSE environment. If you use CMT using the update sets procedure, then this restriction does not apply.

## Maintenance version

To be able to run a merge between different BSE environments you must use a so called Update Set. To create such an Update Set you must create a Maintenance version. This maintenance version registers and displays versions you adjusted and want to merge. The sources of these adjusted versions must be placed in the Update Set. You cannot create an Update Set without the Maintenance version.

## Update set

An Update Set is another name for a set of sources, which are changed for a modification in the software. This modification can be a bug fix or a customization. Each source in this Update Set has two versions. The first version is an original source. The second source is the source with a modification built in.

To be able to use CMT for applying modifications to a third version, an original version and a modified version is saved. The original and modified sources are used to apply the modification to a third version.

Before you can start the actual merge you must save somewhere an original version and a modified version. This lists the sessions you need:

- **Maintenance Versions** (ttcmt1100m000)
- **Merge Code by User** (ttcmt1101m000)
- **Update Sets** (ttcmt1110m000)
- **Connect Scripts to Update Set** (ttcmt1210m000)
- **Update Script using Update Set** (ttcmt1220m000)
- **Replicate Maintenance Versions** (ttcmt1200m000)

## Create a maintenance version

1. Start the **Maintenance Version** (ttcmt1100m000) session.

   A Maintenance Version represents a version of the Infor LN or Baan software, which has to be maintained. It can contain different versions of the standard software, localizations, extensions or customizations. Examples are: 41b, B5bsch, B6abcl. This session is needed so you can record Update Sets.

2. Start the**Merge Code by User** (ttcmt1101m000) session.

   This session is to register a Merge Code for each user. CMT needs a Merge code to be able to merge scripts. This Merge code is created automatically during the update of scripts. Also, this code is needed to prevent that different update actions (using the same Merge Code) will disturb each other.

3. Start the **Update Sets** (ttcmt1110m000) session.

   Update Sets contain sources, which have been modified, typically for one modification (customization or bug fix). These sources are connected to the Update Set. When a source has been fixed in a version, an Update Set must be created before the modified version can be copied over the original version. Only Update Sets of Maintenance versions of the current environment are shown.

4. Start the **Connect Scripts to Update Set** (ttcmt1210m000) session.

   This session saves the two versions of the sources in the “Directory for Update Sets”.

5. Start the **Update Script using Update Set** (ttcmt1220m000) will automatically build a modification in another version. This session can only run if the Update Set is created from the Maintenance version where the modification is created initially. Also a Merge code by user must contain a record for the current user that wants to run the merge.

   This ends the process of running a merge with an Update Set. If you want to replicate Update Sets between different BSE environments you can use the session **Replicate Maintenance Versions**.

# Replicate maintenance versions

The **Replicate Maintenance Versions** (ttcmt1200m000) enables you to copy Update Sets between different BSE environments. Normally table data cannot be shared between BSE environments, that is why this session is created. The **Replicate Maintenance Versions** sessions creates replicas of data. Also physical changes for Maintenance Versions of the current environment can be reflected in the tables.