Infor Enterprise Server Administration Guide

10.7.x

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

Publication Information

Release: Infor LN 10.8.x
Publication Date: May 3, 2021
Document code: ln_10.8.x_refesag_op_en-us

Contents

About this Guide

This document explains the main functions and objectives for the administration of LN, and provides information about how the system administration relates to other parts of LN.

Infor Enterprise Server is an application that is used to configure and maintain LN.

Intended audience

This document is intended for the system administrators who set up, configure, and manage the LN software.

The Administration guide describes how to use Infor Enterprise Server to set up an LN environment with LN users, user authorizations, database authorizations, devices, and so on.

You can use this document as a Reference Guide.

Note:

- Some graphics in this document are from previous LN releases and can differ slightly in appearance to your LN sessions. The functionality described is identical.
- This guide contains references to Web UI and Worktop. Web UI and Worktop are only supported if Enterprise Server is used in combination with LN 10.3 or earlier:
  - LN versions earlier than 10.3 support Web UI and Worktop. The HTML5-based LN UI is not supported in these versions.
  - LN 10.3 supports LN UI, Web UI, and Worktop.
  - LN 10.4 and later support only LN UI. Web UI and Worktop are not supported in these versions.

Prerequisites

To use this document effectively, you require knowledge of:

- The hardware and operating system on which you run LN.
- The Relational Database Management System (RDBMS) you use and the relational database concepts.
- Additional installed software.

Related documents

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

- *Infor Enterprise Server Technical Manual (U8172 US)*

- *Infor LN Installation and Configuration Guide (U9200)*
- *Infor LN Exchange User Guide (U8405 US)*
- *Infor ES Application Service Manager Administration Guide (U7784)*
- *Infor Enterprise Server Web UI - Installation and Configuration Guide*
- *Infor LN Installation Guide (U9498 US)*
- The Tools section in the LN help. This section contains online manuals on various important topics.

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

You can use Infor Enterprise Server to manage and configure LN, a business software solution that includes applications, tools, and an LN Enterprise Modeler.

All of the components work together as a fully integrated system and supports all aspects of a business. You can use this document as a reference guide. For detailed information on the sessions that are used in the procedures, see the session help.

Administration in perspective

The main functions and objectives for the administration of LN are described in this document.

Information is provided about system information in LN and how system administration is related to the other parts of LN. After you complete the LN installation, you must perform several administrative tasks. For example, defining companies, users, devices, and so on in Infor Enterprise Server before you can use LN effectively.

No specific relation exists between the administration procedures and one or more LN modules. In fact, the administration procedures are related to all of LN 's modules. Therefore, the data entered during the administration of LN is common data and is used throughout the entire LN software.

Chapter 2: System management

The tasks and responsibilities of the system administrator are versatile.

This section describes the main functions and objectives for the administration of the LN software. For example, you must define, users, devices, and so on, in Infor Enterprise Server before you can use LN effectively.

LN 's administration facilities are also used, for example, to create jobs, define queries, and to manage the development environment. The administration procedures relate to all LN modules.

Administration of your LN installation

Because LN can have many users, an LN system administrator can be required for managing the package.

On an installed LN system the LN system administrator must perform these tasks:

- System management, which includes administrative tasks.
- Application administration, which includes administrative tasks in the application.

System management and application management differ by company. Large companies, usually have more than one system manager, each with specific responsibilities.

System administrator tasks

The application administration consists of several tasks that the system administrator must perform with care and precision.

Act as contact person

The system administrator is the contact person between a company and LN. The application administrator is responsible to find a solution for the questions and problems brought forward by the LN users. If required, this task is performed by contacting one of the Global Support Centers.

Have extensive knowledge about the application environment

The system administrator must know in which directories the LN application resides. For example, to make a backup or to rebuild tables, the system administrator must know the exact location of the files and programs involved.

Reorganize tables

The system administrator must frequently reorganize the database tables because the disk space of deleted records is not immediately given back to the system. In other words, although many records in a table are deleted, the size of the table itself does not become smaller. Only after the table is reorganized the disk space of the deleted records becomes available on the system. Reorganizing the database tables becomes important when many records in tables are deleted.

Control disk space

The system administrator must regularly check the disk space on the system because a lack of disk space can cause application errors.

Remove temporary files

LN can create several temporary files that, after use, are no longer useful. To save disk space, the system administrator must remove the temporary files on a regular basis.

Create back-up files

To prevent a loss of data, the system administrator must create back-up files on a regular basis. The reason to make back-up files is so if a file is damaged, the original file can be retrieved from the back-up files.

Control user management

Before a user can work with LN, the system administrator must first define the user data. To avoid the unauthorized use of programs, user authorizations are defined in roles that can be linked to the individual users.

Install updates

If LN sends an update of the LN software, the system administrator must install the update on the system as soon as possible.

Log changes in the system

The system administrator must log the changes to the system. An administrator can change many aspects in the system, for example, the printer definition files. The logged information can be useful for problem solving.

Perform all initialization procedures

The system administrator must know all initialization procedures, for example:

- How to create a new company number
- How to enter data for a new company number
- How to copy an existing company number to another company number. See the Copy companies (ttaad1231m000) session or the To copy a multicompany structure topic in the “Enterprise Server” section in the Infor Web Help.

Create and customize software

To optimize the use of LN, the system administrator can create or customize various software components, for example, menus, forms, and reports.

Perform periodic application runs

Some application programs must run on a regular basis. The system administrator is responsible to run these programs, for example, to update statistics and to print and remove stock transactions.

Maintain the system log

Data about the system and the application is stored in a system log.

The system data includes information about:

- The application environment
- Any reorganization of tables
- The available free disk space
- The back-up schedule
- The user data reports
- The updates
- Changes in the system
- The error messages and solutions

Analyze problems

The system administrator must analyze any problems that occur during the usage of LN.

LN offers various logging features that are helpful for problem analysis, such as logging for database locking.

Logging for database locking

LN offers a logging mechanism to gather statistics on database locking. You can use the statistics to identify long waiting times, long locks, and retries, which cause performance problems.

To use this logging mechanism:

1 To activate the logging, start the **Logging Parameters (ttaad4153m000)** session and specify this information:

- A threshold, in milliseconds, for logging database lock actions.
- A file to store the locking statistics.
- A time frame in which the logging takes place.

After you specified the logging parameters, you must convert them to runtime.

2 To analyze the logging statistics, use the **Generate lock graph html output (ttstplocklog)** session. The session first generates a snapshot file, based on the locking log file(s). This file contains all, or a subset of, the log records. Then, the session generates an html report that displays the contents of this snapshot file.

For details, see the online help of the sessions mentioned.

Chapter 3: License management and validation

A license management mechanism is a copy protection mechanism used to regulate the commercial use of LN.

The unprotected software must be validated and patched before a specified expiration date.

After installation, the core LN application can run for 30 days without a license mechanism. Integrations do not work.

Users can request a temporary key for demo and trial versions. This key is valid for 14 days. The temporary license is mainly introduced to overcome delays in delivery of permanent license keys.

For testing and demonstration purposes, unprotected software is dispatched with limited validity.

To license and validate LN

You validate and license your LN software with the Infor Solution License Manager (SLM). The SLM is a stand-alone product capable of licensing various Infor products.

If you use SLM to validate several Infor products, we recommend that you install at least two SLM servers on two separate systems, combined as one logical SLM server.

Licensing solution

SLM is the central license manager for the majority of the Infor products. SLM is a central license manager in the sense that one license manager can provide licenses to a variety of products. Installing and configuring a dedicated license manager individually by product is not required.

To establish a licensing solution, you require these components:

- The Infor application or product that has adopted SLM licensing
- The Solution License Manager product itself for handling the licenses. The License Manager checks the license requests from the adopting applications, according to the information that is stored and validated with Infor.
- The Activation key that enables SLM. Submission of the Activation key enables the licenses that your adopting applications require to work.

SLM license types

Infor applications can be licensed in various ways. To see how your application must be licensed, see the Software License and Support Agreement (SLSA). Depending on the Infor pricing strategy, a

restricted set of license types can be assigned to a specific application. For example, some applications can be licensed through any license type, where others can only be licensed through a server license.

In general, the license types are distinguished in node locking and user locking. With a node-locking license, the adopting application can only work on a specific node in your network. A network node can be a server but also a desktop. With a user-locking license, the adopting application can only work with specific named users or with a limited amount of concurrent users.

Currently, SLM supports these license types:

- Concurrent User license

  A user-locking license in which a pool of concurrent users can use the adopting application.
- Named User license

  A user-locking license in which a pool of specific named users can use the adopting application.
- Server license

  A node-locking license in which a pool of specific named server or desktop computers can use the adopting application.
- Instance license

  An instance-locking license in which a pool of specific named instances can be used for the adopting application.

  Note: With SLM 2.5, instance licensing is registered by means of server licensing.
- Desktop license

  A node-locking license in which a set of adopting applications are linked to a pool of specific named server or desktop computers.

After a license request is made, the SLM server checks the license configuration (License file). This file contains the information on how many desktops, users, servers, or instances can use the application. Then, according to the license type, the specific named user, server, or desktop is verified.

Some adopting application can be licensed with all the various license types, where others can handle only a selection of licenses.

Named User license

If an adopting application uses Named User licensing, the SLSA states the number of named users that can use that application. The SLM server keeps track of a list of all specific users that are permitted to use that application. The server rejects a license request from an adopting application if the SLM server does not know the specific user name.

Infor validates the amount of named users and never knows the specific names of these users.

Some applications automatically synchronize the user names of their authorization module with the SLM server.

The SLM server checks for a specific user name in combination with the desktop name where the application is running. You can specify that a specific user can run the application concurrently from multiple desktops. Each desktop on which this user runs the application decreases the number of licenses by one.

License Administration in LN

In case of a Named User License, you can use the License Administration sessions in LN to keep the Named User list in SLM up to date.

A short description of the License Administration sessions:

- **SLM Product IDs by User (ttslm0130m000)**

  Use this session to display or edit the SLM Product IDs that are used by a user.

- **Link SLM Product IDs to Users (ttslm0230m000)**

  Use this session to link SLM Product IDs to Users that are based on the Authorization Management System. This session links the product IDs to users, based on which sessions the users are authorized for.

- **Synchronize User Data with SLM (ttslm0230m100)**

  With this session you can synchronize user names with the SLM for product IDs that require a Named User License.

Chapter 4: System recovery

This section describes the relevant procedures and steps to carry out the hard-crash recovery procedure of an LN Software Environment (BSE) on UNIX and Windows platforms.

After a hard-crash, you must restore your LN Software Environment (BSE) from a back-up file stored on another system. Therefore, before you use your BSE in an operational environment, you must plan a back-up strategy for your BSE.

Back-up strategy

Before you can use LN, you must decide how you plan to protect your BSE against potential disk crashes and other system failures. If you do not plan and implement a correct back-up procedure, you cannot recover your BSE after a system crash.

You must thoroughly test your back-up and recovery procedure before and after you move to a run-time environment. You must perform test recoveries on a regular basis to ensure that your archiving and backup procedures work.

To ensure a successful recovery scheme, frequent backups are essential. You must base the back-up frequency on the frequency of changes in your BSE's data. For example, if data changes at a high rate, the backup frequency must be proportionally high.

The amount of time you must keep the back-up files depends on the back-up frequency of your recovery scheme.

For additional safety, maintain two or more of your previous backup files, in case part of the current backup file is lost or damaged.

System recovery procedure

- Procedure aim

  To identify the steps that are required in a hard-crash procedure if you cannot reboot your operating system and restart your BSE.
- Procedure result

  This procedure results in a new BSE that functions the same as the initial installation.
- Prerequisites

  You must have on another system an undamaged backup version of the files that have crashed in your BSE environment.

Procedure summary

To recover LN:

1. Restore the back-up files.
2. Start and, if required, configure the Infor Solution License Manager (SLM).
3. Start the printer daemon and the shared memory:
   - On a UNIX platform, run the `rc.start` shell script
   - On a Windows platform use the Infor ES Service Manager
4. Start the BSE environment.

In addition, you might also be required to recover your operating system and database. See your operating system and database manuals for the appropriate procedures.

Note: Important information, such as database parameters, login accounts, Kernel parameters, and so on, must be logged in a safe place. For example, to reuse this information during a recovery, log information in a system or database administrator's log book.

Chapter 5: Version and release management

This section describes LN 's version and release management, which manages the various versions of the LN packages, their corresponding releases, and customizations.

Version and release management overview

You must manage various versions of packages, corresponding releases, and all the various customizations on the standard software.

Infor Enterprise Server offers a comprehensive solution with a version and release management concept. The advantages of the version and release concept are:

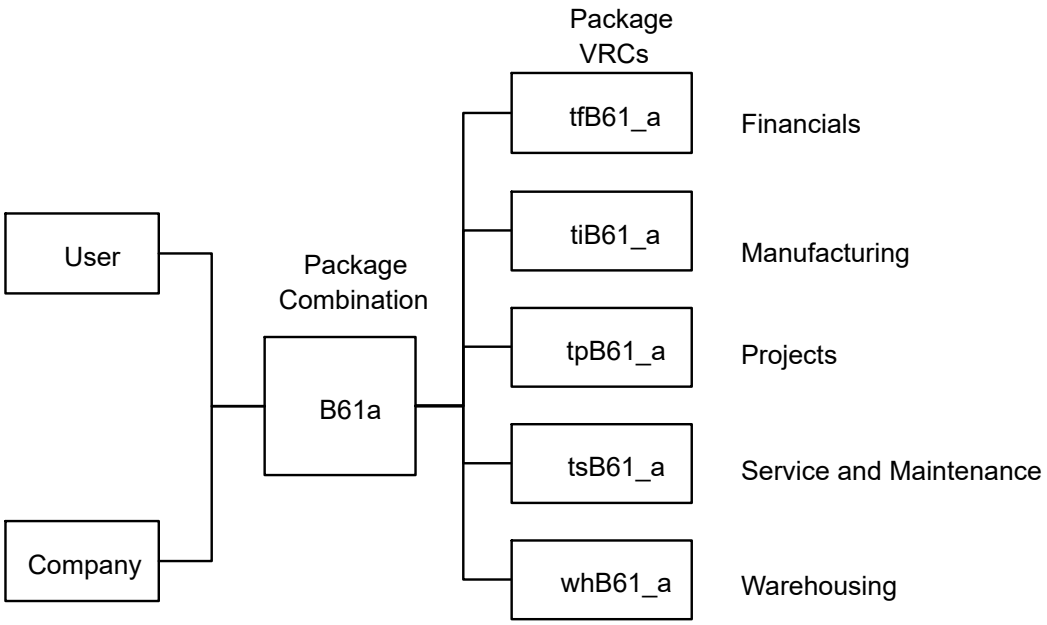
- A flexible development environment
- Flexible management of the software, such as:
  - New versions
  - Patches
  - Customizations
  - Localizations
- Developers can create and test the customizations on the standard software in a separate development environment.
- Operational users do not see new or customized software components until the components are fully tested and released in their own run-time environment.

Package combinations

A *package combination* is a collection of various LN packages. Every package combination has a unique Version Release Customer (VRC) identity.

The package combination links users to a specific version of the LN software. A package combination can be linked to one or more users and to one or more companies. A package combination can contain only one version of a package, which is identified by a package VRC.

This diagram shows an example of a standard package combination:



This example standard package combination contains several standard package VRCs, such as the standard version of LN Financials.

Package VRCs

Each LN package is distributed as a *package VRC*, which represents a specific version of that package.

VRC stands for version release customer code, and it identifies the development stage of the LN software.

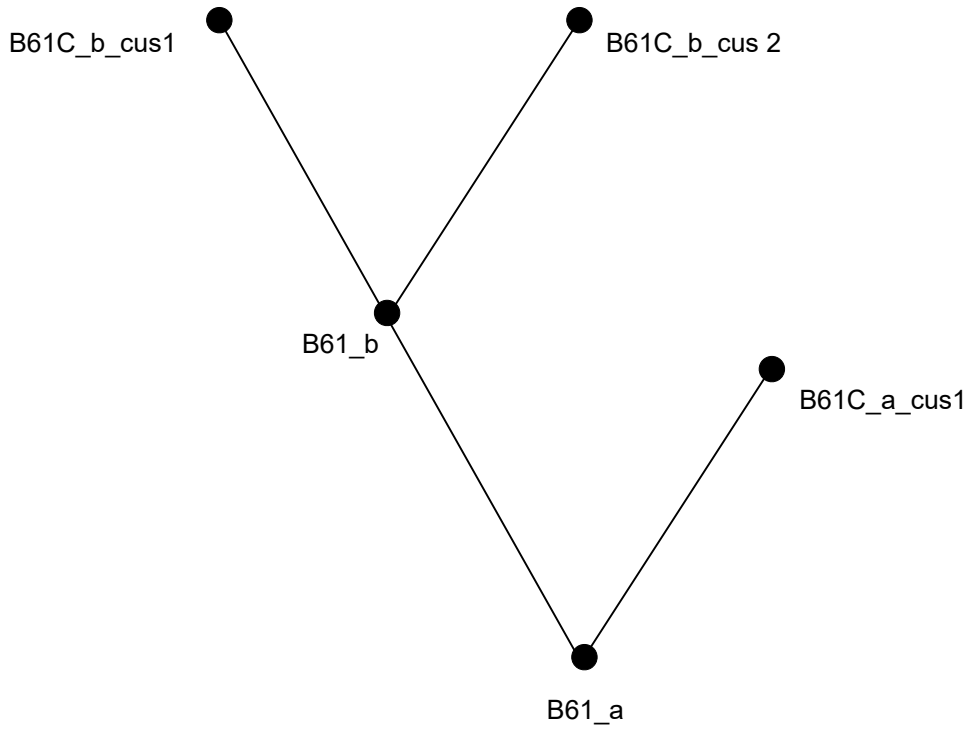
A VRC consists of these elements:

- A version code that identifies a major LN software modification.
- A release code that identifies a minor LN software modification.
- A customer extension code that identifies LN software for a specific customer.

A package, for example, LN Financials, can have more than one version. Each version is identified by a specific package VRC. For example, LN Financials can have a package VRC that contains the standard software of LN Financials and a package VRC that contains the customizations on the standard software of LN Financials.

As a rule, a new package VRC is derived from a previous package VRC. To avoid duplication of software components, a new VRC only contains those software components that have changed compared to previous VRCs.

This diagram shows an example of the VRC derivation structure:



In this example, B61_a is the VRC that contains the standard software. These new VRCs are derived from this VRC:

- The VRC B61C_a_cus1 contains a customized version of the standard software
- The VRC B61_b contains a new version of the standard software. From this new standard VRC, two new customized versions are derived:
  - B61C_b_cus1
  - B61C_b_cus2.

  The VRCs B61C_b_cus1 and B61C_b_cus2 contain several customizations on the B61_b standard version.

Package VRC code

A unique identifier identifies package VRCs in LN. LN uses the pp vvvt_rr_gggn format to identify package VRCs.

The codes are described in this table:

Code	Description
pp	Package code
vvv	Version code

Code	Description
t	Type code. Denotes customizations on standard software. This field is not filled for the standard software. Possible type codes include: • L Localizations: Customizations for a specific country. • B Branch: Customizations for a specific line of business. • C Customer: Customizations ordered by the customer, developed by a dealer or partner. • O Own: Customizations developed by the customer.
rr	Release code
ggg	Customer code: This field is not filled for the standard software.
n	Sequence code

For example, for the Strangely Brown Chicken company (SBC), the package VRC tc B61O_b_sbc1 represents the first customization on the standard software in the B61_b VRC of LN Common.

Note: Customers can only create new VRCs for their own customizations from the standard software. Infor or a partner delivers all other VRC types.

LN software environment

To create specific software environments, you can use *package combinations*.

A software environment has these components:

- A package combination
- At least one company

You can create a special package combination to customize software components or to create new software components.

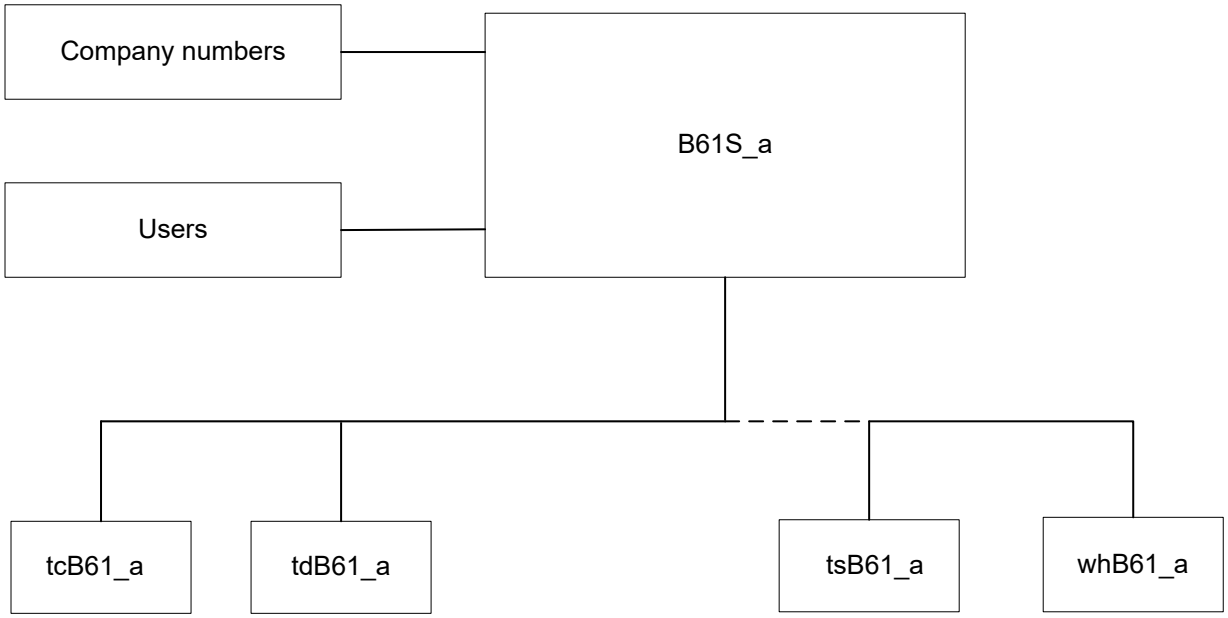
This table shows the software environments in LN:

Software environment	Description
Standard environment	A package combination that contains the standard version of all LN packages.
Runtime environment	A package combination that contains the package VRCs that normal users use at runtime.
Development environment	A package combination that contains package VRCs in which developers can create customizations.

Standard environment

The standard environment contains the standard software of all LN package VRCs of all LN packages. You can use the standard environment to compare customized software with standard software.

This diagram shows an example of the standard environment.

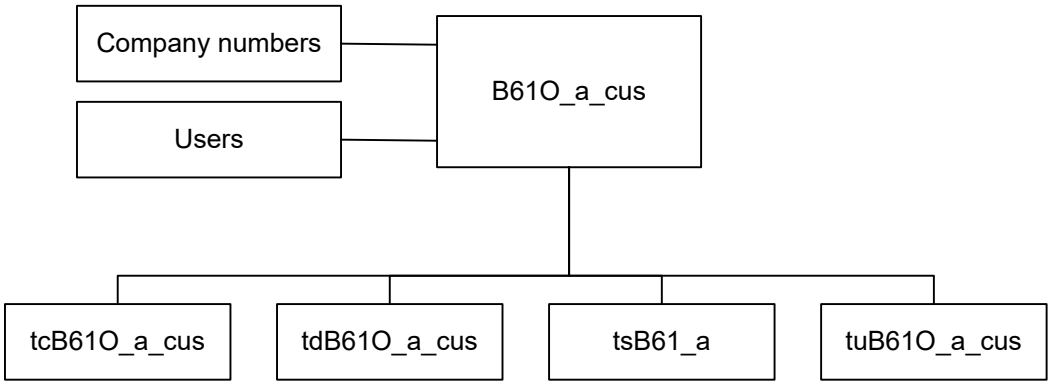


The diagram shows an example of a user who is linked to a specific package combination. Use the User Data (ttaad2500m000) details session to link the user to a package combination.

Runtime environment

A runtime package combination can contain some package VRCs that are derived from standard package VRCs. It is not required that all packages in the package combination have the same package VRC. If no customizations exist for a certain package, the standard VRC of this package is used in the run-time package combination. In the diagram of a typical runtime environment, the Service and Maintenance package (tsB61_a) is not customized. The other packages are customized and for these the derived package VRC is used in the package combination.

To specify that the package combination is customer defined, ensure you add not only the customer extension (cus) but also the extension O. This letter indicates that the package combination is the customer’s own.

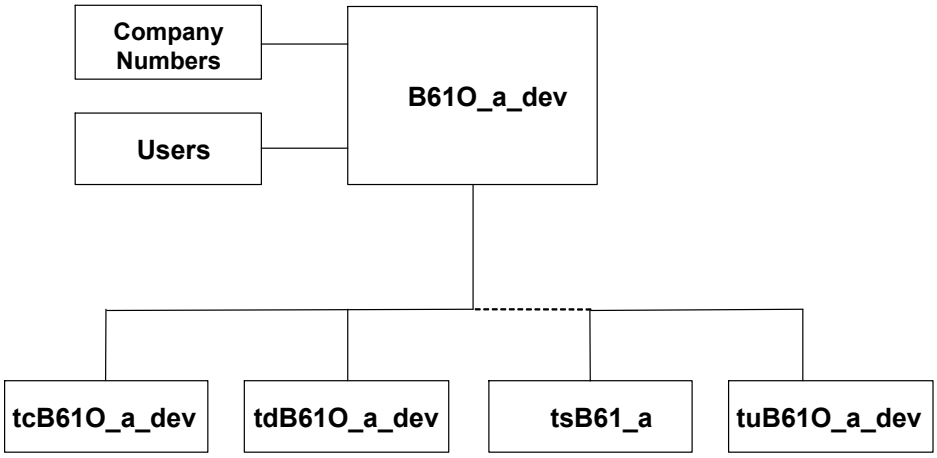


Development environment

As in the run-time environment, a package combination in a development environment contains package VRCs that are derived from standard package VRCs. It is not required that all packages in the package combination have the same package VRC. If no customizations exist for a certain package, the standard VRC of the package is used in the run-time package combination.

Developers use a development environment to develop new software components or to change existing components.

This diagram shows an example of a development environment:



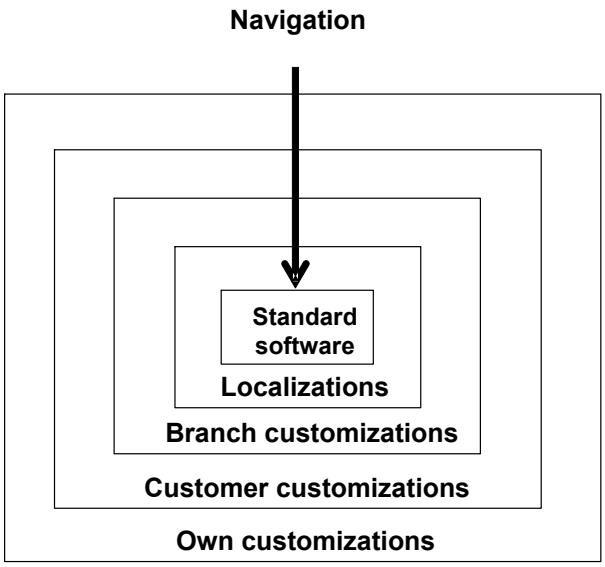
The name of this package combination, such as the previous one, contains the character O. In this case, the customer extension is replaced by the development extension (dev).

VRC derivation

If a user starts a software component, for example a session, the Virtual Machine (VM) searches from the outside to the inside.

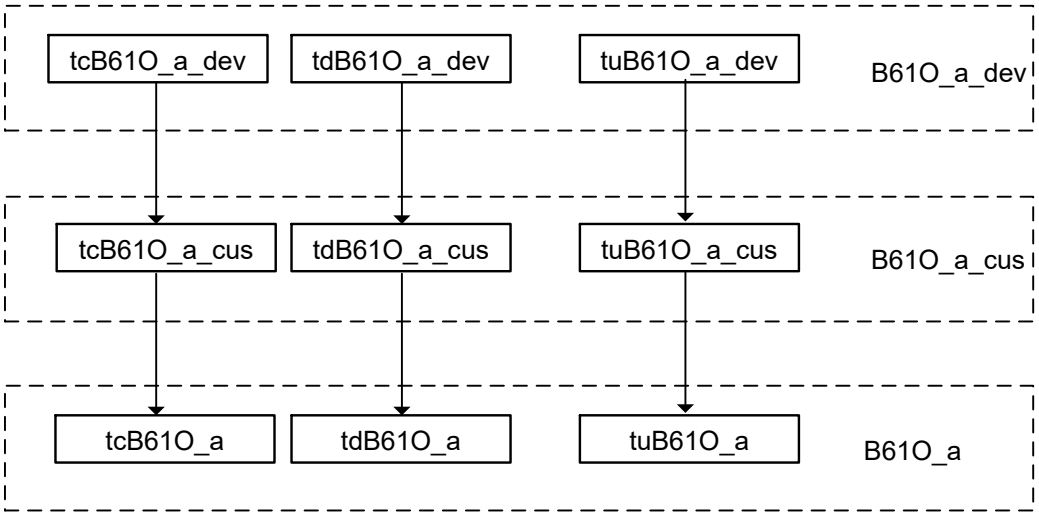
The Virtual Machine (VM) searches for any customizations on the component made by the customer in the own customization. If no own customizations exist, the Virtual Machine (VM) searches for a customer-specific customization, and so on.

This diagram shows how the Virtual Machine (VM) navigates through the customizations on a component. It searches from the outside to the inside.



The search path also depends on the *package combination* and company to which the user is linked. For example, the *package VRCs* in a development package combination are derived from the VRCs in the run-time package combination. Those run-time package combinations were, in turn, derived from standard package VRCs.

This diagram shows the derivation structure of the development package combination.



The VRCs with customer extension dev are derived from the VRCs with the cus extension, which themselves are derived from standard VRCs. If a user works with package combination b61O_a_cus and starts a software component. The Virtual Machine (VM) searches for a customization in the VRC with the cus extension. If such a customization is found, this customized component is started. If no such customization exists, the component in the standard VRC is started.

If a user works with package combination b61O_a_dev and starts a software component. The Virtual Machine (VM) searches for a customization in the VRC with the dev extension. If no such customization is found, the Virtual Machine (VM) continues the search in the cus VRC. If this VRC also does not contain a customized version of the component, the Virtual Machine (VM) starts the component in the standard VRC.

If you create many package VRCs, derived from each other, the derivation structure can become complicated.

A cluttered and complicated VRC derivation structure has these consequences:

- A decrease in system performance due to the longer derivation structure.
- A cluttered directory structure due to a large number of newly created package VRCs.

The maximum VRC depth, in other words, the number of VRC that you can stack on top of each other, is 40 VRCs. For best results, keep the maximum VRC depth as low as possible.

Standard software environment procedure

A software environment is created that you can use to create your own customized software for LN.

The environment contains a dedicated *package combination* and *package VRCs* for a customization department.

Procedure result

A new software environment with package combinations and package VRCs.

Prerequisites

Ensure that you have at least 10 MB free disk space and 40 MB database space available.

This section describes only the most important steps in the standard software environment procedure. For detailed instructions on how to enter data in the sessions that are mentioned in the procedure steps, see the session help.

Note: You can use the procedure on both UNIX and Windows platforms. You can use the ${BSE} notation for both platforms, and for the slash (/) and backslash (\) characters.

Points of attention

Points of attention for the package combination:

- The new package combination must contain a *VRC* for each package, except Tools (tt and tl).
- The Data Director (da), OpenWorld Middleware-enabling (tm), and New Technology (nt) must not be customized. Therefore, you must include existing (not your own!) VRCs for these packages. For example, include the highest update or localization VRC in the derivation structure.
- For the remaining packages (except tt and tl) you must include your own VRCs.

Creating a standard software environment

1. In the Package VRCs (ttadv1511m000) session, create a new package VRC.
2. In the Directories of Software Components (ttadv1115m000) session, create new directories for the software components of the new package VRCs.
3. In the Package Combinations (ttaad1520m000) session, create a new package combination.
4. In the Packages by Package Combination (ttaad1121m000) session, link the package VRCs to the new package combination.
5. In the Change Package Combination by Company (ttaad1101m000) session, link companies to the new package combination.
6. Change Package Combinations for Users (ttaad2200m000) link users to new package combination.
7. In the Developer Authorization Template (ttams1151m000) session, change the authorizations for the developers.
8. Optionally, use the **Initialize Tools (tlcom0200m000)** session to initialize the Test tool for Business Data Entities (BDEs).
9. In the User Data (ttaad2500m000) session, authorize the normal users for a new or other environment.

One-step software environment

A software environment is created that you can use to create your own customized software for LN.

The environment contains a dedicated *package combination* and *package VRCs*

You cannot use this procedure to maintain existing environments. To maintain existing environments, you must use the sessions that are discussed in the for a customization department.Standard software environment procedure on page 27. The prerequisites for and the result of this procedure are identical to those of the Standard software environment procedure.

Procedure result

A new software environment with package combinations and package VRCs.

Prerequisites

Ensure that you have at least 10 MB free disk space and 40 MB database space available.

This section describes only the most important steps in the one-step procedure. For detailed instructions on how to enter data in the sessions that are mentioned in the procedure steps, see the session help.

Note: You can use the procedure on both UNIX and Windows platforms. You can use the `${BSE}` notation for both platforms, and for the slash (/) and backslash (\) characters.

Points of attention

Points of attention for the package combination:

- The new package combination must contain a *VRC* for each package, except Tools (tt and tl).
- for a customizationDo not customize Data Director (da), OpenWorld Middleware-enabling (tm), and New Technology (nt). Include existing (not your own!) VRCs for these packages. For example, include the highest update or localization VRC in the derivation structure.
- For the remaining packages (except tt and tl) you must include your own VRCs.

Creating a one-step software environment

1. In the Create New Package Combination / VRCs (One Step) (ttaad1222m000) session, create the software environment.
2. Optionally, with the **Initialize Tools (tlcom0200m000)** session, you can initialize Test tool for Business Data Entities (BDEs).

Chapter 6: LN Software Maintenance

To help you manage software updates to your LN system, Infor offers the Product Maintenance and Control (PMC) Tool. An efficient, highly effective tool for managing functional *software updates* (Feature Packs) and other software updates (Individual Solutions).

The PMC module manages the installation of *Feature Packs* and *Individual Solutions*. The PMC module is delivered to all customers as part of the master CD-ROM of LN. The PMC module is used to check software updates for completeness and *customization* interference.

The PMC module contains two sub-modules.

- *PMC distributor*
- *PMC recipient*

PMC Benefits

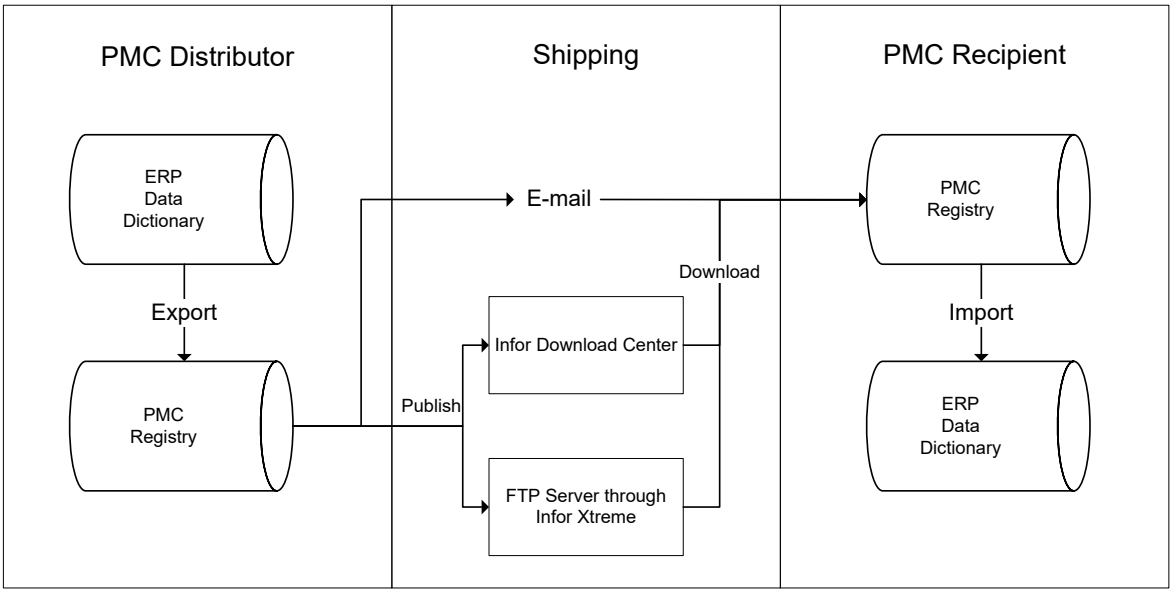
The PMC module is a tool that enables you to install, configure, and run new software. The PMC module simultaneously addresses a wide range of software update challenges, including:

- Automatic checks for updates for any previous *dependencies*
- Analysis of potential conflicts with system customizations at installation time. This enables the user to identify any conflicting customizations, to ensure that these exist on the system on which you install the update.
- Option to maintain copies of all previous software components. This feature enables the user to roll back updates to any point in history.
- Overview of all Feature Packs and Individual Solutions, that are installed.

Note: A customization component does not have a ‘maintenance date’ in its object. Therefore, if you deliver customization components to a customer, whose maintenance license has expired, the customer can still deploy the components.

PMC Architecture

This diagram shows the architecture of the PMC module:



The software developer creates or updates software components in the Data Dictionary of the developer's ERP system. Software is packaged by the *PMC distributor* and stored in the PMC Registry. Feature packs are published on the Infor Download Center. Individual solutions and monthly collections are published on an FTP server. Software can also be sent by email.

The customer can download feature packs from the Infor Download Center. Through Infor Support Portal, the customer can download individual solutions and monthly collections from the FTP Server. *PMC recipient* builds or updates the PMC registry on the customer's ERP system. Software in the customer's PMC Registry can be imported in the customer's data dictionary.

Where to find the PMC module

The PMC module is part of Infor Enterprise Server (Tools). The software supplier is the distributor of the solutions, and the user or customer of the applications is the recipient. The PMC module contains functionality and sessions for both the distributor and recipient role.

To navigate to the recipient part of the PMC module:

1. Log on to LN.
2. On the ERP Menu Browser, select **Tools > Software Installation > Miscellaneous > Recipient**.
3. Select **Setup > Operational > Miscellaneous**.

Alternatively, to navigate to the recipient part of the PMC module:

1. Log on to LN.
2. On the ERP Menu Browser, select **Tools > Application Development > Product Maintenance and Control > Recipient**.
3. Select **Setup > Operational > Miscellaneous**.

To navigate to the distributor part of the PMC module:

1. Log on to LN.
2. On the ERP Menu Browser, select **Tools > Application Development > Product Maintenance and Control > Distributor**.
3. Select **Setup > Operational**.

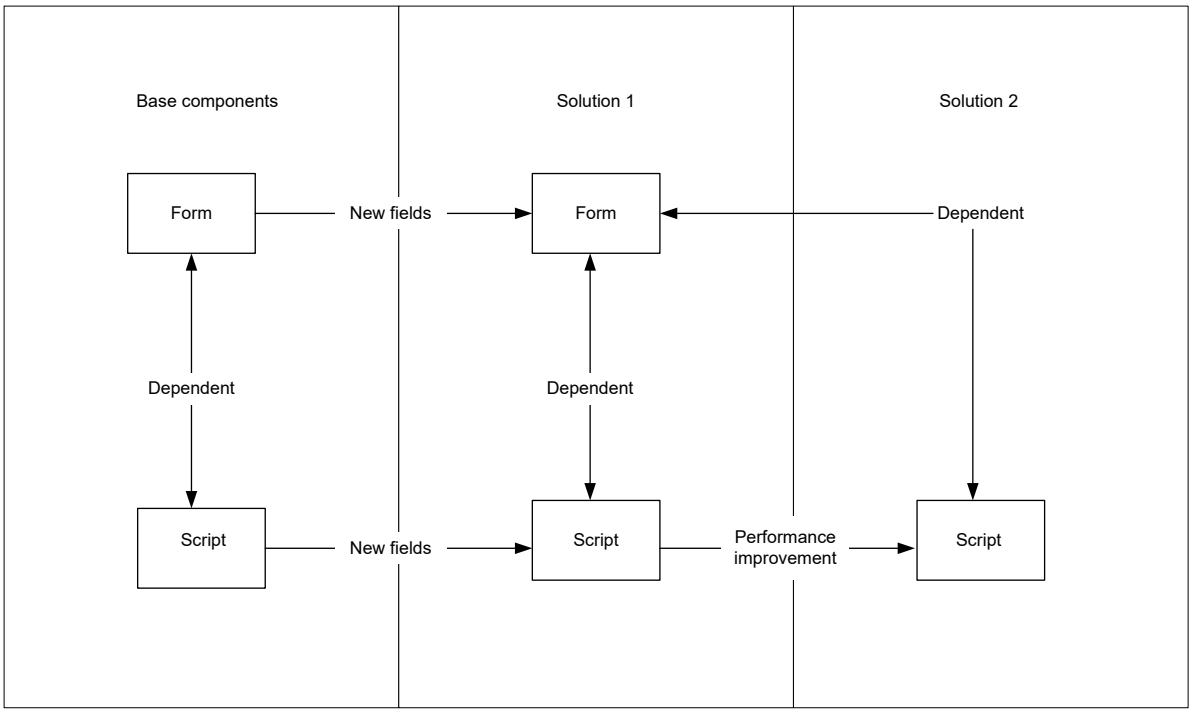
Dependencies

When the PMC recipient installs software updates, the recipient checks the presence of the required depending solutions.

These dependency types exist:

- *Pre-requisites*
- *Co-requisites*
- *Post-requisites*

This diagram shows a prerequisite dependency: solution 1 must be installed before solution 2.



Example

A dependency requires you to install the previous version of a software component first. You have installed LN from the master CD-ROM without customizations or solutions. You receive an updated version of the **Copy Sales Order (tdsls4201s000)** session. This session contains the session object, script, and forms that are interdependent of each other.

You decide not to install the dump. Later you discover that a change is required in the script to correct an issue that is keeping your sales force from selling an item. The change is sent as a dump that

contains only the session object and script. You can install this dump if you installed the previous version of the software component. The dump requires the correct form of the first solution. PMC automatically installs the previous solution.

Suppose you deliberately decided not to install a solution. For example, because you do not know if the existing version of the **Copy Sales Order (tdsls4201s000)** session was customized. The PMC module provides protection against this. In this case PMC informs you that the current version is customized. You cannot overwrite software, because PMC uses update VRC levels and maintains copies of all previous software components. With this setup you can roll back updates to any point in history.

Individual solutions

A PMC Distributor creates and publishes *individual solutions*.

Individual solutions that are created for standard products of Infor are available for download 24 hours a day, seven days a week. See the Infor Support website http://infor.com/support for Infor-owned products.

You can only download these solutions one by one.

After you download and scan a solution, the PMC recipient checks the dependencies with other solutions. Sometimes, other dependent solutions might be missing at the installation system and must be downloaded and scanned, as well. The process to install individual solutions can take a long time if many dependent solutions must be downloaded and scanned.

You can use the Multilevel Download functionality to download all the dependent solutions, without user interaction, to simplify and speed up the download process.

Collections

Collection dumps enable customers to update their system with the latest software.

Collections are groups of all individual solutions that are released in a particular period.

These individual solutions are gathered at the PMC distributor side in a single dump file.

Collections for standard Infor products are published on an FTP server.

You can scan collections in a single action. With the Scan Solution/Patch File (ttpmc2200s000) session, the dump file is split into all individual solutions that were included in the collection.

Afterwards, you can install the individual solutions.

You can use collection dumps in two ways: for pro-active installation or corrective installation.

- Pro-active installation

  The user installs all individual solutions of the collection. The user runs the most up-to-date version of the software, including the latest released fixes.
- Corrective installation

  The individual solutions of the collection are not installed. The solutions have status Available in the PMC registry. If the user runs into a problem, and a solution is advised for this problem, the user can install that solution immediately. All dependent solutions are installed automatically.

Feature Packs and patches

Periodically, individual solutions can be bundled into *Feature Packs* or patches. In general, these types of updates contain solutions created in a larger time period than a collection. In the PMC Tool, the term ‘patch’ is used both for Feature Packs and patches. The patch entity is known at both the PMC distributor and PMC recipient side. Feature Packs and patches are an indivisible set of solutions. You cannot install or uninstall individual solutions that belong to a Feature Pack or a patch by the PMC recipient. You can only install or uninstall entire Feature Packs and patches. Dependencies between Feature Packs and patches can exist.

The subsequent sections describe the justification, characteristics, and differences of Feature Packs and patches.

Justification of Feature Packs and patches

During the life cycle of a product, in general, two types of changes are implemented in the product:

- Corrective fixes for defects.
- Functional enhancements to further enrich the product.

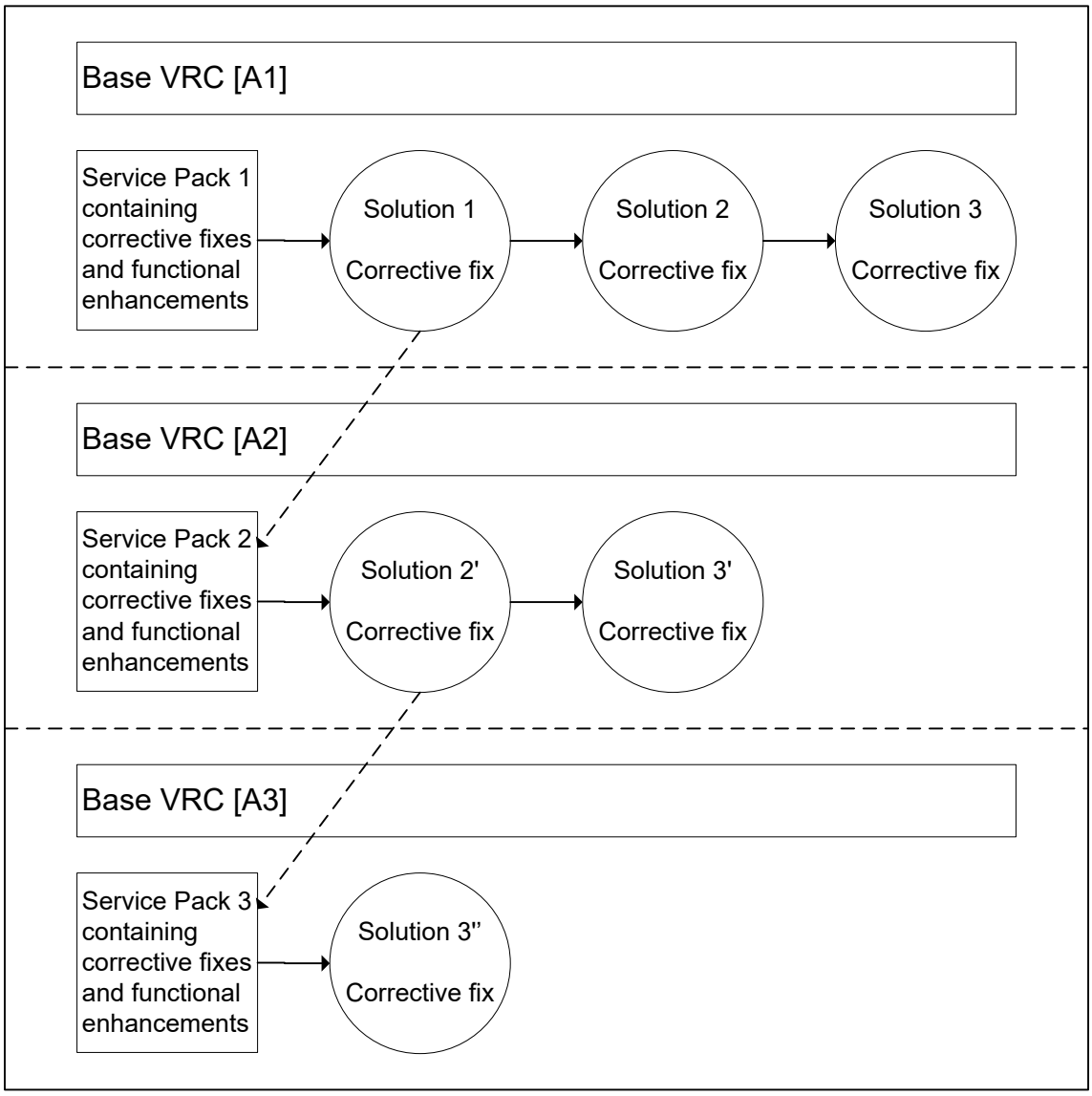
Functional enhancements are often accompanied by changes in the data model, changes in the user interface, and changes in the behavior of the application. In general, functional enhancements require a careful preparation before installation in an operational environment. PMC generates dependencies between solutions, that contain the same components. This is to guarantee that the solutions are installed in the correct order and no required solutions are skipped. Sometimes, users must first install several functional enhancements before they can install a corrective fix for a relative small defect. Installation of the functional enhancements can be required at unexpected and undesirable moments.

With Feature Packs you can separate functional enhancements from the flow of corrective fixes. This separation enables you to adopt and implement functional enhancements in a more controlled way. Patches do not offer this possibility.

Feature Packs

With *Feature Packs* you can separate corrective fixes from functional enhancements. Multiple parallel maintenance baselines exist for the product. Every new baseline is linked to a unique *base VRC*. Functional enhancements are, in principle, only implemented at the start of a new baseline.

This diagram shows the Feature Packs mechanism:



Feature Pack 1 is linked to base VRC A1. The Feature Pack contains several corrective fixes and some functional enhancements. Corrective maintenance is performed for this Feature Pack, which results in solutions 1, 2, and 3 in the previous diagram.

In parallel, the software distributor develops Feature Pack 2, which is linked to base VRC A2. Whenever possible, the distributor immediately ports and includes corrective solutions created on top of Feature Pack 1 in Feature Pack 2. See solution 1 and the dashed arrow in the previous diagram. These ported solutions are registered in Feature Pack 2 as so-called obsolete solutions.

The same process repeats when the distributor develops Feature Pack 3, which is linked to base VRC A3.

After the release of Feature Pack 2 and 3, the distributor also creates corrective solutions on top of these Feature Packs.

When installing Feature Pack 2 at a recipient, the PMC Tool checks whether solutions that were installed on top of Feature Pack 1 are also available for Feature Pack 2. These solutions can be available in two different shapes for Feature Pack 2:

- As an *obsolete solution*.

  Included in Feature Pack 2. See solution 1 and the dashed arrow in the previous diagram. These obsolete solutions are integrated in Feature Pack 2 and do not have a separate physical solution dump.

- As an *individual solution*.

  Available on top of Feature Pack. See solutions 2’ and 3’ in the previous diagram. These so-called missing solutions are not included in Feature Pack 2 and do have a physical solution dump.

  Note: When Feature Pack 1 is installed in multiple update VRCs. The PMC Tool performs the missing solution check in all update VRCs, that contain Feature Pack 1.

The PMC Tools refuses to install Feature Pack 2 if not all solutions that are installed on top of Feature Pack 1 are available for Feature Pack 2. Otherwise, old defects can return after you install Feature Pack 2. The distributor must ensure that all solutions of the preceding Feature Pack are available for the subsequent Feature Pack before publishing the subsequent Feature Pack.

Note: You can change this default behavior, so that you can install Feature Pack updates without additional individual solutions. See "To install a 'clean' Feature Pack".

Feature Packs are cumulative. Every new Feature Pack also includes all preceding Feature Packs. This enables you to skip the installation of Feature Packs at the PMC recipient. For example, you can immediately upgrade from Feature Pack 1 to Feature Pack 3 without installing Feature Pack 2 in between. The installation process is optimized in such a way that only components that are changed as compared to the current installed Feature Pack are installed.

You can install Feature Packs in an existing update VRC. Therefore, creating a new update VRC is unnecessary. Therefore, the VRC structure at a recipient system can remain unchanged. The *base VRC* that is linked to the *update VRC* changes. You can also install a new Feature Pack in a new update VRC.

To install a 'clean' Feature Pack

You can update an existing environment with a Feature Pack through PMC or through the Installation Wizard. It is required that all the solutions, that were installed on top of a previous Feature Pack, are installed for the new Feature Pack too. PMC requires these additional solutions to prevent that old issues return after installation of the Feature Pack. This applies also if you install the Feature Pack in another, or even in a new, Update VRC. This is because you can upgrade your companies to the Package Combination of that Update VRC and then you might see old issues again.

In some situations this functionality is too strict, for example when you want to develop customizations for different customers on different Feature Pack levels.

To change this default behavior, you must set the PMC_IGNORE_SOL environment variable to 1. You can now install a 'clean' Feature Pack, without taking care of the content of already installed Feature Packs and additional solutions.

You can specify the environment variable in different ways:

- In the command field of the BW configuration file (-set PMC_IGNORE_SOL=1). In this way the setting only applies to the user that uses this configuration.
- In the $BSE/lib/bse_vars file (add the line PMC_IGNORE_SOL=1). If you use the Installation Wizard you can add this line when you are prompted to change the bse_vars configuration file. In this way the setting applies to all users.

Note:

- We recommend that you remove the PMC_IGNORE_SOL setting after each Feature Pack installation.
- When installing the Feature Pack in an existing Update VRC, the installed solutions in that Update VRC are also installed for the new Feature Pack. To ignore additional solutions only applies to solutions of other Update VRCs.

Example

You want to install Feature Pack 2, which includes solutions 133, 146 and 154 as obsolete solutions already.

For some Update VRCs with different Feature Packs and additional installed solutions, this table shows what happens in these situations:

- PMC_IGNORE_SOL is not set.
- PMC_IGNORE_SOL is set.

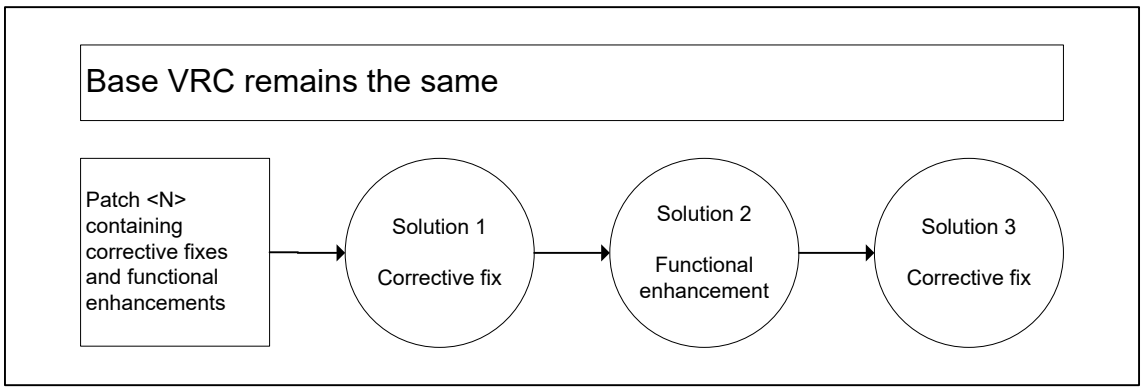
	Update VRC		
	B61U_a_stnd	B61U_a_fp1	B61U_a_fp2
Feature Pack that is installed	-	FP1 including obsolete solutions 133 and 146	-
Solutions that are installed	133 146 154 168	154 168 172	-
Action	Install FP2 in an 'FP0' Update VRC with some individual installed solutions.	Install FP2 in an Update VRC with FP1 and some additional installed solutions.	Install FP2 in a new Update VRC.

	Update VRC		
Result if PMC_IGNORE_SOL is not set	Fails unless additional solutions 168 and 172 are present. Reason for failure: Solution 168 was already installed in this Update VRC, and solution 172 was installed on top of FP1 in Update VRC B61U_a_fp1.	Fails unless additional solutions 168 and 172 are present. Reason for failure: Both solutions were already installed in this Update VRC.	Fails unless additional solutions 168 and 172 are present. Reason for failure: Solutions 168 and 172 were installed on top of FP1 in Update VRC B61U_a_fp1.
Result if PMC_IGNORE_SOL is set to 1	Fails unless additional solution 168 is present. Reason for failure: Solution 168 was installed in this Update VRC, so it is needed on top of FP2. Solution 172 can be ignored now.	Fails unless additional solutions 168 and 172 are present. Reason for failure: Both solutions were installed in this Update VRC, so they cannot be ignored.	Succeeds. Reason for success: PMC can ignore all solutions, because it is a new Update VRC.

Patches

Patches do not enable you to separate corrective fixes from functional enhancements. Only one maintenance baseline of the product exists. Changes are always implemented in the latest version of a software component. The base VRC remains the same. Functional enhancements are included in the chain of depending corrective solutions.

This diagram shows the Patch mechanism:



To install solution 3 in this diagram, you must first install solution 2, which contains a functional enhancement.

Patches are not cumulative, which means you cannot skip the installation of Patches at a recipient's location. For example, before you install Patch 3, you always must install Patch 2 in advance. You cannot immediately upgrade from Patch 1 to Patch 3.

Distributor's Policy

The software distributor decides if Feature Packs or Patches are being applied. A mix of both types is also possible:

- When applying Feature Packs, the distributor can release a functional enhancement as an individual PMC solution on top of one of the released Feature Packs.
- When applying Feature Packs, the distributor can decide to periodically also release Patches on top of the Feature Pack.

When applying Feature Packs, the distributor's policy determines how many base VRCs are maintained in parallel and how long the base VRCs are maintained.

PMC Distributor functionality

The software developer generates additional software and makes software changes on the existing LN software packages.

The supplier of the LN software uses the PMC Distributor functionality to manage the software updates and prepares those updates for delivery to the customers.

This list shows a process overview of the PMC Distributor module:

- Create individual solutions:
  1. Define a unique identifier for the *solution* and a brief description.
  2. Link one or more software components to the solution.
  3. Define or generate the *dependencies* on other solutions.
  4. Export the solution, the software dumps are now created.
  5. Release the solution, the solution is now ready for delivery.
- Create collections:
  1. Define a unique identifier for the *collection* and a brief description.
  2. Link one or more individual solutions to the collection.
  3. Export the collection.
  4. Release the collection.
- Create patches:
  1. Define a unique identifier for the *patch* and a brief description.
  2. Link one or more individual solutions to the patch.
  3. Define or generate the dependencies on other patches.
  4. Validate the patch.
  5. Export the patch.
  6. Release the patch.
- Create Feature Packs:
  1. Define a new base VRC for the *Feature Pack*.
  2. Define a unique identifier for the Feature Pack and a brief description.

3. Define the dependency on previous Feature Packs.
4. Generate the Feature Pack, in other words, link all components in a specified VRC to the Feature Pack.
5. Validate the Feature Pack.
6. Export the Feature Pack.
7. Release the Feature Pack.

- Additional distributor functionality:
  - Mutilevel export.
  - *SCM* integration.
  - Maintenance History.

PMC Recipient functionality

Customers use the PMC Recipient module to install the LN software updates on their ERP system.

An overview of the PMC Recipient module.

- Download solutions

  You can download solutions from the FTP server or copy the solutions from a medium such as a CD-ROM.
- Scan and connect solutions

  The solution files can be scanned. In other words, the user must extract and store the solutions into the PMC registry. The extracted solutions are also connected to an update VRC.
- Process solutions
  1. Check to install. Report this: Pre- and post-installation instructions, customized components, and missing dependent solutions.
  2. Install the solutions: Store the software components in the data dictionary, report additional post-installation instructions.
- Additional recipient functionality
  - Uninstall
  - Mutilevel download
  - Compare installed solutions
  - Solution History
  - PMC cleanup
  - Copy PMC registry
  - View installation runs.

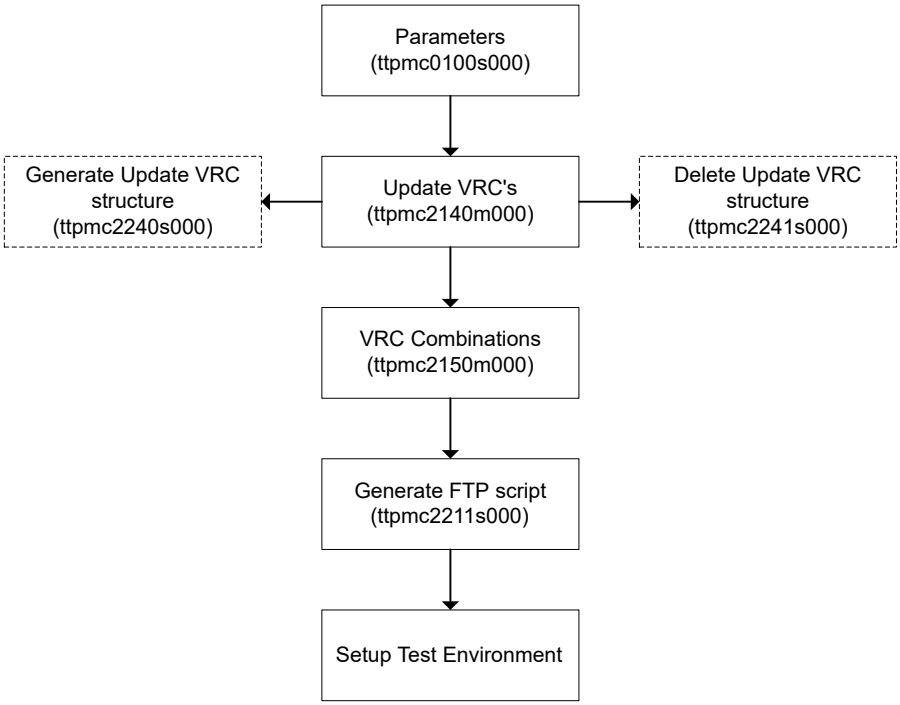
PMC recipient procedure

PMC recipient is fully configured by the Infor Installation Wizard during the installation of the base Infor products such as LN and Infor Enterprise Server.

The setup and how to make changes to the setup are described.

Procedure

This diagram shows the recipient setup procedure steps:



Infor Installation Wizard and PMC

The PMC Tool has a seamless integration with the Infor Installation Wizard. If you install *updates* using the Infor Installation Wizard, the wizard activates the PMC Tool in the background.

You can use both the PMC Tool and the Infor Installation Wizard to install updates in an existing environment. Both tools serve their own specific goal in the update installation process.

The Infor Installation Wizard is primarily for installing these components:

- Initial master media
- Feature Packs

The PMC Tool is primarily for installing these components:

- Individual solutions
- Collections
- Patches

You can also install individual solutions, collections, and patches using the Infor Installation Wizard. This can have these drawbacks:

- The installation process is more labor intensive, more mouse clicks.

- You can miss essential information during the installation process. The so-called pre installation and post installation instructions are not presented in an active way.
- The Infor Installation Wizard installs all solutions present in the staging area or server. You cannot select a subset of solutions from the available solutions.

For these reasons, to install individual solutions, collections and patches, ensure to use the PMC Tool directly.

On the other hand, you can install Feature Packs directly using the PMC Tool. After you have installed the Feature Pack you must also perform several additional steps manually. These steps include running the Convert to Runtime Data Dictionary (ttadv5215m000) , Compile Labels (ttadv1243m000) and Create Tables (ttaad4230m000) sessions. If you use the Infor Installation Wizard, all required steps are executed automatically in the installation process. In addition, the Infor Installation Wizard offers some extra features that can be useful during installation of Feature Packs. For example, the automatic creation of new VRC structures and creation of Package Combinations in which the Feature Pack can be installed.

You cannot perform the initial installation of initial master media using the PMC Tool. To perform this installation, you must use the Infor Installation Wizard. During installation of the master media, the Infor Installation Wizard initializes the PMC Tool. The environment is immediately ready for installing updates. You can also initialize the PMC Tool manually with the PMC Tool itself.

Parameters

The first step in the Recipient setup procedure is to define the PMC parameters.

These parameter groups are available:

- Recipient

  The *PMC recipient* part of the PMC module uses these parameters and these parameters must be filled.

  The parameters are paths on the operating system, on which the solution dumps are stored.

- Distributor

  Even if you do not use the *PMC distributor* part of the PMC module, you must fill the parameters of this group for technical reasons.

See the Parameters (ttpmc0100s000) session.

Note:

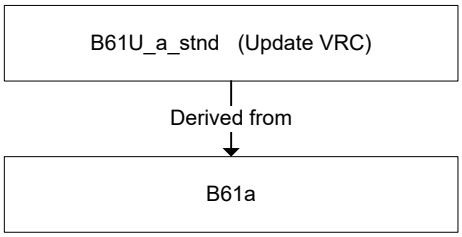
- Because parameter definition is a one-time step, you must be aware of the effect if you change the parameters afterwards.
- If you change the operating system paths, you must move the directories and files according to the new path specifications.
- If you have more than one ERP environment running, for example, a Production and a Test environment. You can share the solution dump directories on the ERP environments, which reduces disk-space consumption.

Update VRCs

PMC uses *update VRCs*. The software of an initial standard master product is installed in a root VRC that is predefined by the software supplier. The update VRC is created on top of this root VRC. Feature Packs and individual solutions that you download from the Infor Support website https://www.infor.com/support are also installed in this update VRC. The update VRCs are created automatically during the installation of the master product.

You can use the Generate Update VRC Structure (ttpmc2240s000) session to generate these update VRCs. In addition, you can define update VRCs using the Update VRC's (ttpmc2140m000) session.

This diagram shows the VRC structure for standard LN:



B61_a is the VRC that contains the baseline, which is the first shipment of LN 6.1. From this point on, Individual solutions and Service Packs are installed in the B61U_a_stnd VRC.

For the Infor Enterprise Server (packages tt, tl, tm, da and nt) no new update VRCs are created and used.

After a fresh installation of LN, these update VRCs are present in PMC:

The heading of the first column in the table, **Pack.**, refers to the package to determine the VRC structure.

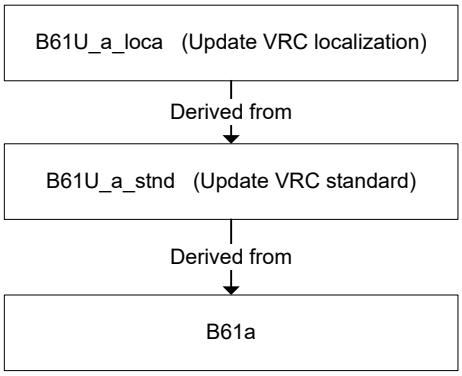
Pack	Update VRC			Base VRC		
da	3.3U	b	stnd	3.3	b	da
tt	7.6	a		7.6	a	tt
tf	B61U	a	stnd	B61	a	

VRC combinations

In PMC you use a *VRC combination* to control *co-requisite* dependencies between solutions of various base VRCs.

Update VRCs that must be updated simultaneously, for example, an update VRC for standard LN and an update VRC for a localization, must be grouped in a *VRC combination*. Use the VRC Combinations (ttpmc2150m000) session and the Update VRC's (ttpmc2140m000) session.

This diagram shows a VRC structure, including a localization:



The table shows the possible Update VRCs in LN when applying a localization.

The heading of the first column in the table, **Pack.**, refers to the package to determine the VRC structure.

Pack	Update VRC			Base VRC			VRC Comb.
da	3.3U	b	stnd	3.3	b	da	
tt	7.6	a		7.6	a	tt	
tf	B61U	a	stnd	B61	a		UPD
tf	B61U	a	loca	B61L	a	loca	UPD

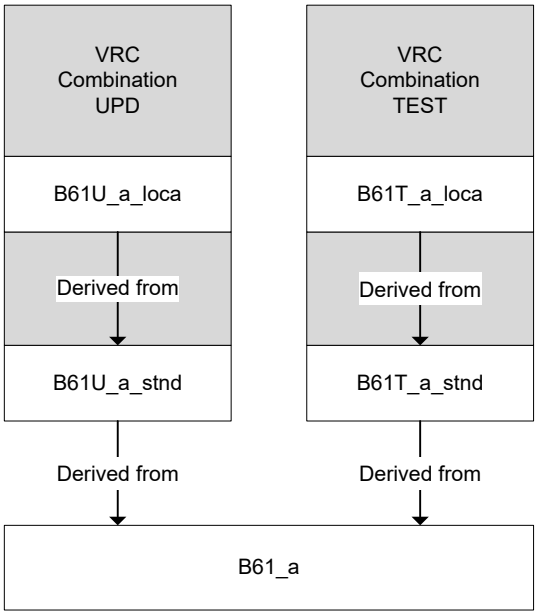
To create a test environment

Best practice is to test the software updates in a test environment before you implement the updates in your live environment.

To test the software updates, create the update VRCs and the VRC combination for your test environment.

Note: For the Infor Enterprise Server (Tools), no update VRC is created. Therefore, you cannot create an update VRC for testing purposes. To test the Infor Enterprise Server software updates before installation in your live environment, you require a separate installation of LN.

Create the update VRC for a test environment parallel to the VRCs in your live environment. This diagram shows an example:



The update VRCs that are grouped in a VRC combination, for example, B61U_a_stnd and B61U_a_loca are updated simultaneously.

The B61T_a_stnd and the B61T_a_loca are updated in the same way. These VRCs represent a test environment.

Note: If you do not have a localization or extension installed, no VRC combination is required. The only update VRC you require for your test environment is B61T_a_stnd.

To create a PMC test environment:

1 Start the Generate Update VRC Structure (ttpmc2240s000) session. Specify this information:

- In the **Package to Determine VRC Structure** field specify the package **tf**. Do not use any other packages, such as **cp**.
- You can choose any customer VRC extension for the **Update Customer** field. Note that the **Update Customer** fields are the five fields on the far right in the **VRC Information** group box. We recommend that you use **stnd** for standard VRCs and the same value as the base VRC customer code for localizations and extensions. If you do not have a localization installed, leave the **Localizations** fields blank.
- Specify the **VRC Combination** field and a description. This step is not required if you do not have a localization installed.
- The **Customization Type** field can contain any character, but the character **T** is recommended for a test.
- Select the **Create Update VRCs** check box.
- Clear the **Change Derivation of Other VRCs** check box to specify that you do not want to derive the current VRCs from the new update VRC.
- Clear the **Update Package Combinations** check box to specify that you do not want to change the VRCs in package combinations.

2 If you do not yet have a package combination that contains the new VRC structure, you must create a package combination. Use the Package Combinations (ttaad1520m000) and Packages by Package Combination (ttaad1121m000) sessions to set up a package combination.

3 Start the Create Runtime Data Dictionary (ttadv5210m000) session.

Specify the information in the dialog box. Click **Create Runtime** for all package combinations in the range of updated package combinations, you selected in the Generate Update VRC Structure (ttpmc2240s000) session.

Select these options:

- Domains
- Tables
- Check before create runt.
- Reconfigure tables
- Sessions
- Print Errors

FTP/HTTP to download software updates

If the dumps of the solutions to install are located on an FTP/web server, you can use the **Download, Scan and Connect Solution (Multi-Level) (ttpmc2210s000)** session to download the dumps.

Before you start the download, ensure the correct download settings are specified.

To specify the download settings:

1 Start the **Parameters (ttpmc0100s000)** session.

2 Under **Download Settings**, select HTTP or an FTP download.

3 If you selected the HTTP protocol, specify this information:

- **Download Site**: The web server where the solutions can be found.

  Default = `secure2.support.baan.com`
- **Download User**: Your login name for this server.

  Default = your current login name including the domain. For example, `infor\<Loginname>`
- **Download Password**: The password required to log on to the web server.
- **Download Directory**: The folder where the solutions can be found.

  Default = `/ftpdownload/updates/`, note the starting and ending slash.

4 If you selected the FTP protocol, specify this information:

- **Download Site**: The ftp server where the solutions can be found.

  Default = `ftp://ftp.support.baan.com/`
- **Download User**: Your login name for this server.

  Default = your current login name including the domain. For example, `infor\<Loginname>`
- **Download Password**: The password required to log on to the ftp site.
- **Download Directory**: The folder where the solutions can be found.

  Default = `/updates/`, note the starting and ending slash.

5 To test the settings, click **Test Connection**.

6 If you selected the FTP protocol, optionally generate an FTP script by clicking **Generate FTP Script**. The name of the generated FTP script is `ftpbaan`. The script is stored in the `$BSE/lib` directory on the ERP server.

The **Download, Scan and Connect Solution (Multi-Level) (ttpmc2210s000)** session uses either method to make the connection to the FTP/HTTP server and to download the solution dumps.

To install updates

This section describes the procedures for the recipient of updates to manage the PMC administration.

Procedure steps

This section provides a summary of the procedure steps to install and uninstall software updates through PMC.

To install updates:

1 Download updates.
2 Scan updates.
3 Connect update to update VRC.
4 Check update to install.
5 Install update.

To uninstall updates:

1 Check update to uninstall.
2 Uninstall update.

Download updates

To download updates you can use the Download, Scan and Connect Solution (Multi-Level) (ttpmc2210s000) session. The session automatically scans the downloaded solutions, and connects the solutions to the specified update VRC. The purpose of scanning the solutions is to make the solutions known to the system, and to identify the installation conditions. After the solutions are scanned, you might find that the solutions cannot be installed unless other solutions are downloaded and installed as well. In that case, the session also downloads, scans and connects these solutions. This step is repeated, until all required solutions are downloaded, scanned, and connected. If the related solutions are already present on your system, but connected to another VRC, the solutions are not downloaded again. They are only connected to the specified update VRC.

You can also download solutions manually, in which case you must also complete some steps. These steps are integrated in the Download, Scan and Connect Solution (Multi-Level) (ttpmc2210s000) session, and can be skipped if you downloaded the solutions using this session.

Note: You can only use this session for downloading if the solution dumps are published on the FTP server and defined in the Generate FTP Script (ttpmc2211s000) session.

Scan updates

The first step to make the solutions available to the PMC registry is to scan the solution files with the Scan Solution/Patch File (ttpmc2200s000) session. The solutions receive the status **Available** in the update VRC for which the scan is carried out. All the relevant PMC data is stored in the PMC registry.

Connect update to update VRC

Use the Connect Solutions/Patches to VRC (ttpmc2201s000) session to make the solutions available in other update VRCs. You can disconnect the solutions with the Disconnect Solutions/Patches from VRC (ttpmc2202s000) session.

Check to install

All functionality is controlled by the Process Solutions (ttpmc2101m000) session and the Process Patches (ttpmc2102m000) session.

The solutions with the status **Available** Check Solution/patch to Install (ttpmc2203s000) session. The related solutions are checked as well. The solutions, including the related solutions, which must be installed receive the status **To Install**. Optionally, detailed information is given on customized components and components that exist in non-supported languages.

Install

You can install the solutions with status **To Install**, with the Install Solution/Patch (ttpmc2205s000) session. If the PMC registry was changed after the solution received the **To Install** status, the solution must be checked again before the solution can be installed. The related solutions are installed as well. After the solutions are installed, the status of the solutions is and connected to the update VRCs must be checked with the **Installed**.

Check to uninstall

All functionality is controlled by the Process Solutions (ttpmc2101m000) session, the Process Patches (ttpmc2102m000) session and the Installation runs (ttpmc2503m000) session.

If a solution must be uninstalled, for example, because the solution was a bad fix, use the Check Solution/Patch to Uninstall (ttpmc2206s000) session. The solution status and the status of related solutions becomes **To Uninstall**.

Note: You can use **Check to uninstall** from the Installation runs (ttpmc2503m000) session to uninstall all solutions that were installed together in one run. For example, a solution including all prerequisite solutions or solutions of a collection that were installed in a range.

Uninstall

Run the Uninstall Solution/Patch (ttpmc2208s000) session to set the status of the solution, including the related solutions, back to 'Available'.

Note: If during installation or uninstallation the process is stopped, for example, due to a system crash, process kill, and so on. The solution that was being processed at that point still has the status **Installing** or **Uninstalling**. With the next start of a processing session from the Process Solutions

(ttpmc2101m000) and connected to the update VRCs must be session, PMC recognizes this situation. It starts the stopped process again for the interrupted solution. If a sequence of solutions was to be installed or uninstalled, you must restart the process for the remaining solutions.

Update policy

A procedure to install, test and make a new Feature Pack operational in your live environment is described. The procedure minimizes the affect in terms of system downtime for the operational environment.

Scenario

The master product in VRC B61_a and Feature Pack 1 is installed. It includes several individual solutions on top of Feature Pack 1 in update VRC B61U_a_stnd. This update VRC is linked to package combination b61au. Customizations are available in the B61U_a_cus VRC that is linked to the b61cus package combination. Various operational companies are linked to package combination b61cus.

Now you can install Feature Pack 2. But first you want to explore the new Feature Pack in a separate environment without disturbing the operational business.

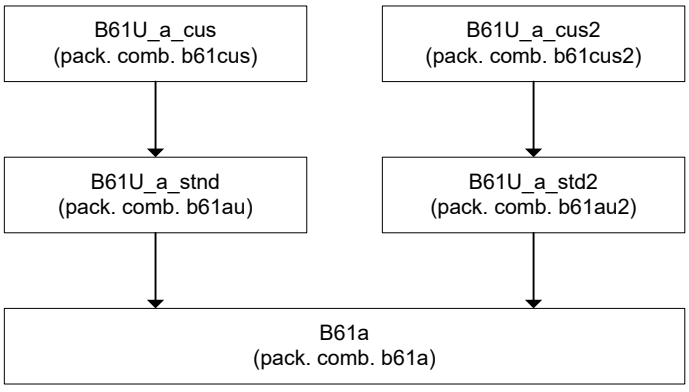
Two methods are available to explore the new Feature Pack in a test environment:

- Method 1

  Switch companies between VRC branches.
- Method 2

  Have a permanent separate test VRC branch.

This diagram shows the recommended update procedure for Feature Packs:



Procedure

Steps 1 through 7 are identical in both methods. Step 8 contains different instructions for method 1 and method 2.

1 Install Feature Pack 2 in the new Update VRC, for example, B61U_a_std2. The new update VRC must be derived from the master product VRC B61_a. In addition, create a new package combination, for example, b61au2. If you install the Feature Pack using the Infor Installation Wizard,

you can have the Infor Installation Wizard create the new update VRC and package combination automatically. The PMC tool guarantees that all individual solutions that were installed on top of Feature Pack 1 in the B61U_a_stnd VRC are also installed in the B61U_a_std2 VRC. Unless these solutions are already included in Feature Pack 2. Therefore, from a maintenance perspective, the B61U_a_stnd and B61U_a_std2 VRCs are synchronous after installation of Feature Pack 2.

2 Companies that are linked to the b61au2 package combination enable you to investigate on runtime the behavior of the standard product without customizations. To achieve this, link one or more test companies to the new package combination b61au2. You can do this in various ways:

- If you have existing test companies linked to the b61au package combination, you can relink these companies to the b61au2 package combination. Use the Change Package Combination by Company (ttaad1101m000) session.
- Copy an existing company that is linked to package combination b61au with the **Copy Company Data (tccom0214m000)** session. Next relink the newly copied company to the b61au2 package combination with the Change Package Combination by Company (ttaad1101m000) session.

3 You can now explore the Feature Pack 2 that is installed in the B61U_a_std2 VRC.

4 Create package VRCs B61U_a_cus2 and the related package combination b61cus2.

5 Copy the contents of the B61U_a_cus VRC to the B61U_a_cus2 VRC using the Copy Software Components to New Package VRC (ttadv6265m000) session. Make the b61cus2 package combination runtime using the Create Runtime Data Dictionary session. Note that, at this point, the customizations are still not synchronized with Feature Pack 2.

6 Copy one or more companies that are linked to package combination b61cus with the **Copy Company Data (tccom0214m000)** session. Next relink the newly copied company to the b61cus2 package combination using the Change Package Combination by Company (ttaad1101m000) session.

7 After you explore the new Feature Pack, upgrade customizations in the B61U_a_cus2 VRC, and prepare your organization to use the new Feature Pack. Verify whether additional solutions were installed in the B61U_a_stnd VRC after you performed Step 1. To perform this verification use the Compare Installed Solutions (ttpmc2405s000) session. Install the missing solutions in the B61U_a_std2 VRC as well. Synchronize the customizations again if required.

8 Depending on the method, method 1 or method 2, perform one of these procedures:

- Method 1: Switch companies between VRC branches:
  - **a** Transfer the operational companies from the b61cus to the b61cus2 package combination. To perform this transfer, use the Change Package Combination by Company (ttaad1101m000) session.
  - **b** After completing the previous steps, all operational companies and users are linked to the b61au2 package combination. To install the subsequent Feature Pack 3, you must repeat the process, but now in the opposite direction: You install Feature Pack 3 in the B61U_a_stnd update VRC. Again, you can create an additional VRC branch for Feature Pack 3. This process requires more installation time and disk space. The test companies are relinked to the b61au package combination. Feature Pack 3 is explored and finally the operational companies are linked to the b61au package combination.

  In theory you can apply the same mechanism for patches. One important difference applies. Patches are not cumulative. Installation of a patch requires that you at least have installed the preceding patch. Therefore, you cannot directly install Patch 2 in the B61U_a_std2 update VRC. You first must install Patch 1, followed by the installation of Patch 2. In the same way, you first must install Patch 2 in update VRC B61U_a_stnd before you can install Patch 3.

- Method 2: Have a permanent separate test VRC branch:

a Install Feature Pack 2 in the B61U_a_stnd VRC. Copy the contents of the B61U_a_cus2 VRC to the B61U_a_cus VRC using the Copy Software Components to New Package VRC (ttadv6265m000) session. Make the b61cus2 package combination runtime using the Create Runtime Data Dictionary session.

b To install the subsequent Feature Pack 3, repeat the process.

In principle you can apply the same procedure for patches when you use Method 2.

Advantages and disadvantages of both methods:

Method 2 has the disadvantage that you must install Feature Pack 2 twice. Therefore the system downtime is greater if you use Method 2.

The system downtime for Method 1 requires time to reconfigure the application database and relink the company and users.

The system downtime for Method 2 requires time for these processes:

- Install and make the Feature Pack runtime.
- Reconfigure the application database.

PMC cleanup

Cleaning up PMC is useful for saving disk space. To run the cleanup start the PMC Cleanup (ttpmc2220m000) session.

This session can clean up three types of PMC data:

- Solution dumps
- Uninstall dumps
- PMC registry

To remove an update VRC

Removing an update VRC can be useful for saving disk space and for a clear VRC structure. To remove an update VRC run the Delete Update VRC Structure (ttpmc2241s000) session.

PMC recipient session summary

This table shows the PMC recipient sessions:

Title	Description
Parameters (ttpmc0100s000)	Maintain the parameters for the PMC module.
Process Solutions (ttpmc2101m000)	Display the *solutions* that are present in the registry of solutions and connected to the specified *update VRC*. Various options are present to process the solutions.

Title	Description
Process Patches (ttpmc2102m000)	Display the *patches* that are present in the registry of solutions and connected to the specified *update VRC*. Various options are present to process the patches.
Update VRC's (ttpmc2140m000)	Maintain the *update VRCs*. This session registers the update VRCs to which solutions can be connected.
VRC Combinations (ttpmc2150m000)	Maintain the *VRC combinations*.
Scan Solution/Patch File (ttpmc2200s000)	Scan an export dump that contains one or more solutions that is placed on the recipient system. You can also use this session to scan *collections* and *patches*.
Connect Solutions/Patches to VRC (ttpmc2201s000)	Connect solutions or *patches* that are already available on the system to an *update VRC*.
Disconnect Solutions/Patches from VRC (ttpmc2202s000)	Disconnect solutions or *patches* from an *update VRC*.
Check Solution/patch to Install (ttpmc2203s000)	Check if the solutions can be installed. The session also signals the problems in regard to customizations.
Check and Install Solutions (ttpmc2204s000)	Check the selected solution or a range of solutions to see whether you can install the solution or range. If all conditions for all solutions in the selected range and their dependency chains comply, the solutions are installed. This session performs the combined functionality from the Check Solution/patch to Install (ttpmc2203s000) and Install Solution/Patch (ttpmc2205s000) sessions.
Install Solution/Patch (ttpmc2205s000)	Install the solution. This process is performed on multilevel. In addition the previous version of the components to install is saved to enable you to uninstall the solution.
Check Solution/Patch to Uninstall (ttpmc2206s000)	Check the solutions to see whether you can uninstall the solutions. You can also use this session to check an *installation run* to be uninstalled.
Uninstall Solution/Patch (ttpmc2208s000)	Uninstall the solution. This process is performed on multilevel. This session is also used to uninstall an *installation run*.
Check Installed Solutions (ttpmc2209s000)	Print the solutions that are installed in an *installation run*.
Download, Scan and Connect Solution (Multi-Level) (ttpmc2210s000)	Download, scan, and connect solutions, including the related solutions. You can also use this session for solutions delivered in a *collection*.

Title	Description
Generate FTP Script (ttpmc2211s000)	Specify some basic data required to download updates from the LN FTP-server.
Login for FTP Server (ttpmc2212s000)	Log on at the LN FTP server to download solutions, *collections*, or *patches*.
PMC Cleanup (ttpmc2220m000)	Clean up PMC. This process helps save disk space.
Generate Update VRC Structure (ttpmc2240s000)	Set up an environment in which PMC works well.
Delete Update VRC Structure (ttpmc2241s000)	Delete an *update VRC* structure.
Copy Solution Registry to Derived Update VRC (ttpmc2290m000)	Copy the solution registry from one *update VRC*, the source VRC, to another update VRC, the target VRC, that is directly derived from that source VRC.
Print Solution/Patch by Update VRC (ttpmc2401m000)	Print detailed information of solutions and *patches* in *update VRCs*.
Compare Installed Solutions (ttpmc2405s000)	List the differences in status between solutions in two *update VRCs*.
Print Dependencies by Update VRC (Multi-level) (ttpmc2440m000)	Print the dependencies between solutions in an *update VRC*.
Print VRC Combinations (ttpmc2450m000)	Print all *update VRCs* for each of the *VRC combinations* in the specified range.
Print Solution/Patch History (ttpmc2460m000)	Print the history of the changes of a solution or *patch*.
Installation runs (ttpmc2503m000)	Display all *installation runs*. From this session, you can use several options on the *solutions* that are installed with an installation run.
Solutions by Installation Run (ttpmc2504m000)	Display solutions by installation run.
History Base VRCs by Update VRC (ttpmc2541m000)	Display history base VRCs by update VRC.
History Solutions by Update VRC (ttpmc2542m000)	Maintain history solutions by update VRC.
Solution History (ttpmc2560m000)	Display the history of the changes of a solution or *patch* in an *update VRC*.
PMC Dependencies Check (ttpmc2441m000)	Generate a report with solutions of an update VRC that is installed when the post-requisites are not yet installed.

PMC distributor procedure

The procedures for the update distributor to set up the PMC administration.

Setup procedure

To perform the setup procedure, run these sessions:

1. **Parameters (ttpmc0100s000)**
2. **Base VRC's (ttpmc0110m000)**
3. **Base VRC Combinations (ttpmc0111m000)**

The subsequent sections provide a detailed explanation of the procedure.

Parameters

The first step in the distributor set-up procedure is to define the PMC parameters.

These group boxes are available:

- Recipient

  Even if you do not use the *PMC recipient* part of the PMC module, you must specify the parameters in this group box.

- Distributor

  The *PMC distributor* part of the PMC module uses these parameters, and therefore you must specify these parameters.

  Most of the parameters are directories on the operating system on which the solution dumps are stored.

See the Parameters (ttpmc0100s000) session.

Note:

Because the definition of the parameters is a once only step, you must be aware of the affect if you change the parameters afterwards.

If you change the directories on the operating system, you must move the directories and files according to the new directory specifications.

base VRCs

You must define a *base VRC* before you can create and link *solutions*, *collections*, *patches*, and *Feature Packs* to the base VRC.

A base VRC is an administrative identifier for a software product. It is not a physical VRC, but must always be connected to a physical VRC, which contains the software components.

The PMC distributor should connect a base VRC to an *export VRC*, the PMC recipient should connect a base VRC to an *update VRC*.

Note: A Feature Pack always starts a new base VRC.

See the Base VRC's (ttpmc0110m000) session.

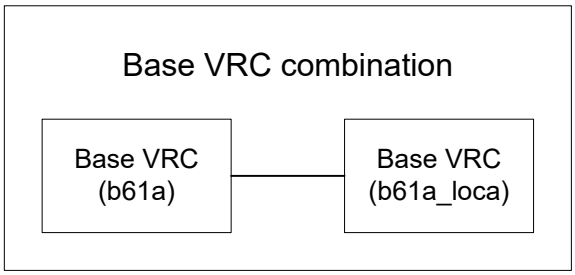
base VRC combinations

Base VRC combinations are required to define relations between base VRCs.

Any base VRC must always be a member of a base VRC combination, even if no relations to other base VRCs exist.

If you have not linked a base VRC to a base VRC combination, you cannot create updates.

This diagram shows a base VRC combination with two base VRCs with a relation:



See the Base VRC Combinations (ttpmc0111m000) session.

To create updates

PMC distributor procedures to create software updates:

- To create solutions
- To create collections
- To create patches
- To create Feature Packs

To create solutions

1 Create the solution.

Create or maintain solutions in the Solutions (ttpmc1100m000) session.

This session is the central session from which you can maintain all aspects of solutions.

The initial *status* of a solution after creation is In Progress.

2 Optionally, create defects and connect defects to the solution.

You can create and connect the defects in the Defects by Solution (ttpmc1110m000) session.

3 Connect software components to the solution.

To connect software components to the solution, use the Components by Solution (ttpmc1520m000) session and the Component by Solution (ttpmc1120s000) session.

You can also connect software components to a solution in the Connect Components to PMC Solution (ttpmc1221m000) session.

4 Define dependencies between the solutions.

To define dependencies between solutions, run the Dependencies (ttpmc1140m000) session.

5 Validate the solution.

To validate a solution, select **Validate Solution** on the *appropriate* menu in the Solutions (ttpmc1100m000) session. This option starts the Validate Solution (ttpmc1404s000) session. This session checks, for example, if components are available, or if components are not compiled in debug mode, and so on.

6 Export the solution.

To export a solution, use the Export Solution/Patch (ttpmc1200s000) session.

7 Release the solution.

To release a solution, you must change the Status from Exported to Released in the Solutions (ttpmc1100m000) session.

You can only release a solution if the Status of the solution is Exported.

8 Optionally, publish the solution.

To publish a solution, select the Published check box in the Solutions (ttpmc1100m000) session.

You can only publish a solution if the Status of the solution is Released.

Optionally, complete these steps:

- View or print solutions

  To view solutions, use the Components by solution (ttpmc1521m000) session.

  To print solutions, use the Print Components by Solution (ttpmc1420m000) session.

- View the solution history.

  To view the solution history, use the Maintenance History (ttpmc1560m000) session.

To create collections

1 Create the collection.

Create or maintain collections in the Collections (ttpmc1503m000) session.

This session is the central session from which you can maintain all aspects of collections.

The initial *status* of a collection after creation is In Progress.

2 Connect solutions to the collection.

You can connect solutions to a collection in the Solutions by Collection (ttpmc1151m000) session.

3 Validate the collection.

Validate collections in the Validate Collection (ttpmc1402s000) session.

4 Export the collection.

Export a collection in the Export Collection (ttpmc1210s000) session.

5 Release the collection.

To release a collection, change the Collection Status from Exported to Released in the Collections (ttpmc1101s000) session.

You can only release a collection if the Collection Status is Exported.

6 Optionally, publish the collection.

To publish a collection, select the Published check box in the Collections (ttpmc1101s000) session.

You can only publish a collection if the Collection Status is Released.

Optionally, view the collection history. To view the collection history, use the Maintenance History (ttpmc1560m000) session.

To create patches

1. Create the patch.

   Create or maintain patches in the Patches (ttpmc1501m000) session.

   This session is the central session from which you can maintain all aspects of patches.

   Clear the New Base VRC check box. Otherwise, the patch is a Feature Pack.

   The initial *status* of a patch after creation is In Progress.

2. Connect solutions to the patch.

   To connect solutions to a patch, you must use either the Solutions (ttpmc1100m000) session or the Base VRC not found session.

3. Define dependencies between the patches.

   To define dependencies between patches, you must use the Dependencies (ttpmc1140m000) session.

4. Validate the patch.

   To validate a patch, you must use the Validate Patch (ttpmc1401s000) session.

5. Export the patch.

   To export a patch, you must use the Export Solution/Patch (ttpmc1200s000) session.

6. Release the patch.

   To release a patch, you must change the Status from Exported to Released in the Patches (ttpmc1501m000) session.

   You can only release a patch if the Status is Exported.

7. Optionally, publish the patch.

   To publish a patch, select the Published check box in the Patches (ttpmc1501m000) session.

   Publish a patch if the Status of the patch is Released.

Optionally, view the patch history. You can view the patch history in the Maintenance History (ttpmc1560m000) session.

To create Feature Packs

Note:

The process to create Feature Packs requires a specific setup of your development environment.

The "Feature Pack development" section describes in detail how the setup must look.

To create Feature Packs:

1. Create the Feature Pack.

   You can create or maintain Feature Packs in the Patches (ttpmc1501m000) session.

   This session is the central session from which you maintain all aspects of Feature Packs.

You must select the New Base VRC check box, otherwise, the patch is not a Feature Pack.

The initial status of a Feature Pack after creation is **In progress**.

2. Define dependencies between the Feature Packs.

   Define dependencies between Feature Packs in the Dependencies (ttpmc1140m000) session.

3. Create the header for the Feature Pack.

   Create the Feature Pack header in the Patch Header (ttpmc1106m000) session.

4. Generate obsolete solutions for the Feature Pack.

   Generate obsolete solutions for the Feature Pack in the Generate Obsolete Solutions (ttpmc1252s000) session.

5. Generate the Feature Pack.

   Generate a Feature Pack in the Generate Patch for New Base VRC (ttpmc1253s000) session.

6. Validate the Feature Pack.

   Validate a Feature Pack in the Validate Patch (ttpmc1401s000) session.

7. Export the Feature Pack.

   Export a Feature Pack in the Export Solution/Patch (ttpmc1200s000) session.

8. Release the Feature Pack.

   To release a Feature Pack, change the Status from **Exported** to **Released** in the Patches (ttpmc1501m000) session.

   You can only release a Feature Pack if the Status of the Feature Pack is **Exported**.

9. Optionally, publish the Feature Pack.

   To publish a Feature Pack, you must select the Published check box in the Patches (ttpmc1501m000) session.

   You can only publish a Feature Pack if the Status of the Feature Pack is **Release**.

Optionally, view the Feature Pack history. You can view the Feature Pack history in the Maintenance History (ttpmc1560m000) session.

Feature Pack development

This section describes various aspects applicable to the Feature Pack development and build process.

System setup for standard products

The subsequent setup assumes that the distributor side includes a combined development/maintenance environment. This combined environment supports the maintenance of already released Feature Packs, development of a new Feature Pack, and the physical creation of solutions and Feature Packs.

Note:

The examples use these conventions for VRC naming:

- Version: B61<X>

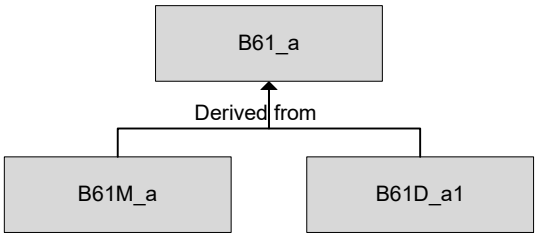
  X = **M** aintenance, **D** evelopment
- Release: a<Y>

Y = Feature Pack number

- Customer: <zzzz>

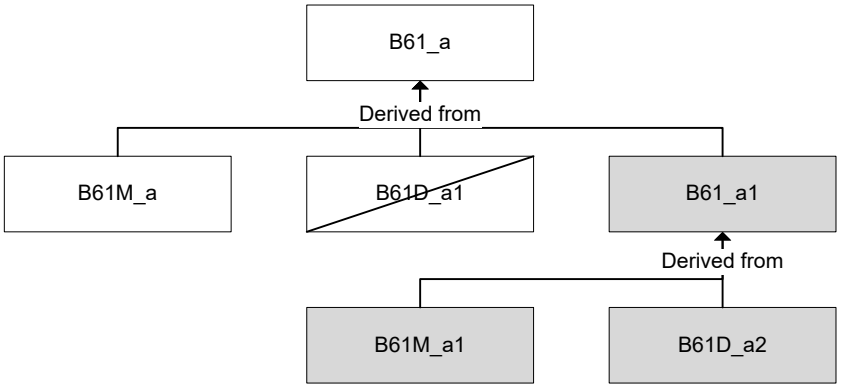
  zzzz = extension/localization code

This diagrams shows the results over time:



When the initial master product, B61_a, which is referred to as Feature Pack 0, is released. The maintenance starts for B61_a in the maintenance VRC B61M_a. The development of the new Feature Pack is carried out in B61D_a1. All fixes in B61M_a are ported to B61D_a1. In the PMC registry of B61D_a1, these fixes are registered as obsolete solutions. After B61D_a1 is frozen, a final Feature Pack is created for the Feature Pack and installed in the same environment. Now you can perform maintenance on the Feature Pack.

Note: The SCM revision data is not available in the Feature Pack VRC B61_a1. The export or import software does not support the transfer of revisions. Because the original development VRC B61D_a1 is still present, you can still view revisions.



Currently, you can maintain B61M_a and B61M_a1, Feature Pack 0 and Feature Pack 1, in parallel. First, port bug fixes that are created for Feature Pack 0 between the freeze moment and the time of installation of Feature Pack 1. Feature Pack 2, B61D_a2, is currently in development.

You can repeat this cycle unlimited times for each new Feature Pack.

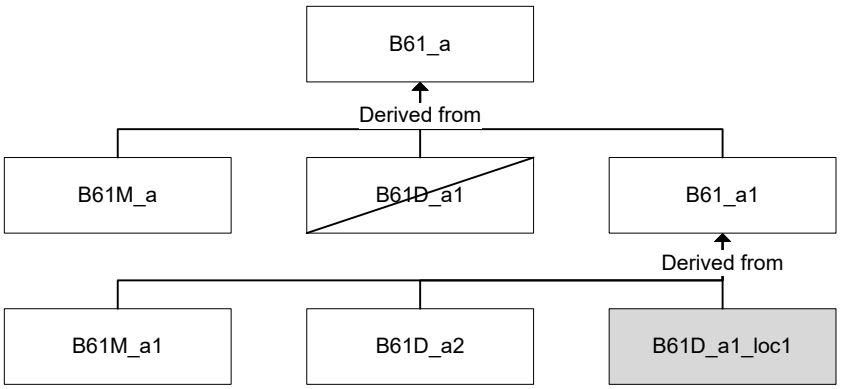
System setup for derived products

The setup becomes a bit more complex if derived products such as localizations or extensions enter the scope.

Note: Customizations are also a type of derived products. For customizations, the situation differs somewhat. The "Customizations" section describes how to deal with customizations.

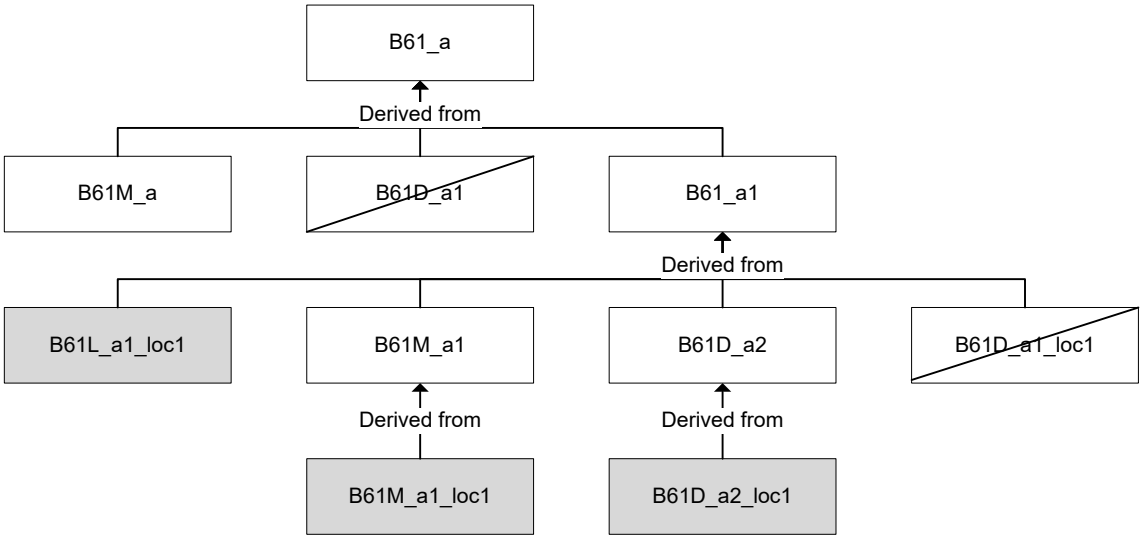
Suppose you must create a localization on top of Feature Pack 1.

A new VRC is created on top of Feature Pack 1, B61D_a1_loc1, as shown in this diagram:



The localization is built on top of the standard Feature Pack and is delivered as a Feature Pack in PMC. Therefore, do not create master media for derived products. As soon as Feature Pack 0 for the localization is available, the Feature Pack is installed in the maintenance VRC, B61M_a1_loc1.

The localization is also copied to a development VRC B61D_a2_loc1 as shown in this diagram:



First you must merge all bug-fixes of the standard VRC B61M_a1 into the localization maintenance VRC B61M_a1_loc1 and create PMC solutions for the localization.

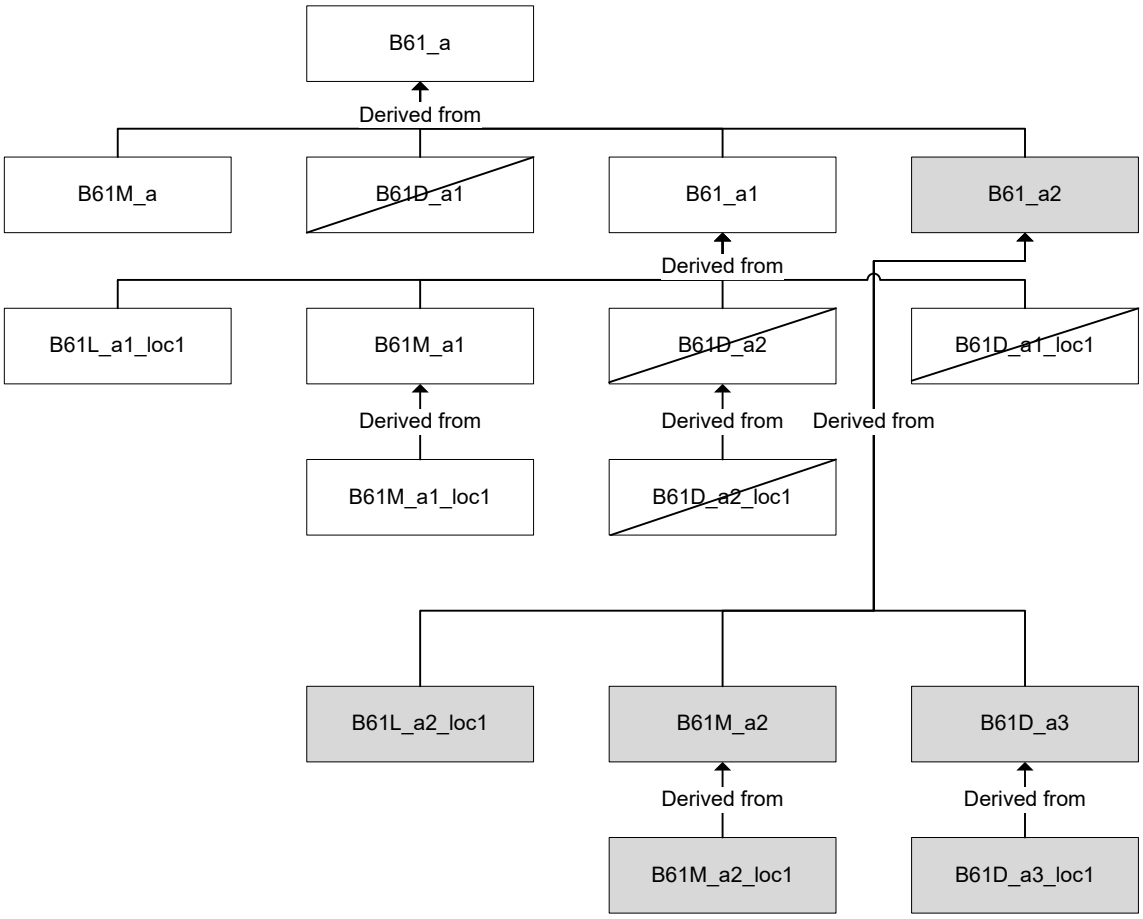
In addition, you must merge the B61D_a2 functional enhancements to B61D_a2_loc1.

From this moment on, the localization follows the same Feature Pack cycle as the standard software.

Repeat the same approach for Feature Pack 2.

The physical Feature Packs for the standard software and the localization are created and installed in a separate VRC tree B61_a2 and B61L_a2_loc1. New maintenance and development VRCs are created.

The maintenance and development VRC for the localization are populated with the components from the installed Feature Pack B61L_a2_loc1. See this diagram:



Rules to upgrade derived products

With respect to the creation of Feature Packs for derived products these rules apply:

- You must only build Feature Packs for derived products on top of Feature Packs of the parent product. Do not build Feature Packs on top of a Feature Pack plus several individual solutions of the parent product. The reason for this is because you can only define PMC dependencies between the Feature Pack of the standard and the derived products. You cannot define PMC dependencies from the Feature Pack of the derived product to individual solutions of the parent product.
- The Feature Pack frequency of derived products can be greater than the Feature Pack frequency of the parent product. Therefore, you can build multiple derived Feature Packs on top of a single Feature Pack of the parent product.
- The Feature Pack frequency of derived products cannot be lower than the Feature Pack frequency of the parent product. If you do not update a derived product in time, various compatibility issues between the parent and the derived product can occur.

Building Feature Packs and the Infor Installation Wizard

PMC Feature Packs are installable units that you can install with the Infor Installation Wizard. Metadata that is included in a Feature Pack controls the behavior of the Infor Installation Wizard. When making your Feature Pack suitable for installation with the Infor Installation Wizard these aspects are relevant:

- Before you build a Feature Pack, ensure that the installable unit key, name, and version are filled at your PMC base VRC. If you do not specify these values, the Infor Installation Wizard does not recognize the installable unit. See the Base VRC's (ttpmc0110m000) session.
- Before you export a Feature Pack, ensure that the Feature Pack header contains the correct properties. Based on the properties that are defined in the Feature Pack header, the Infor Installation Wizard can automatically perform particular actions at a recipient system. For example, create new package combinations, create new update VRCs, and link new update VRCs to an existing or new package combination. See the Patch Header (ttpmc1106m000) session.

Customizations

Various aspects are relevant to the development and maintenance process of customizations.

Customizations are, in most cases, built by organizations other than the producer of the related parent standard product.

The vendor who builds customizations installs the standard parent product in an update VRC.

The customization is built on top of the installed parent standard product.

This situation is specific in the sense that PMC serves both as recipient and distributor in such a development environment.

Patches and customizations

Sometimes, the vendor of the standard parent product applies patches instead of Feature Packs as a method to provide incremental updates to the market. As opposed to Feature Packs, patches are not cumulative. The installation of a patch in an update VRC requires that you also install all preceding update VRCs in the same update VRC. This issue has consequences for the VRC setup at the distributor.

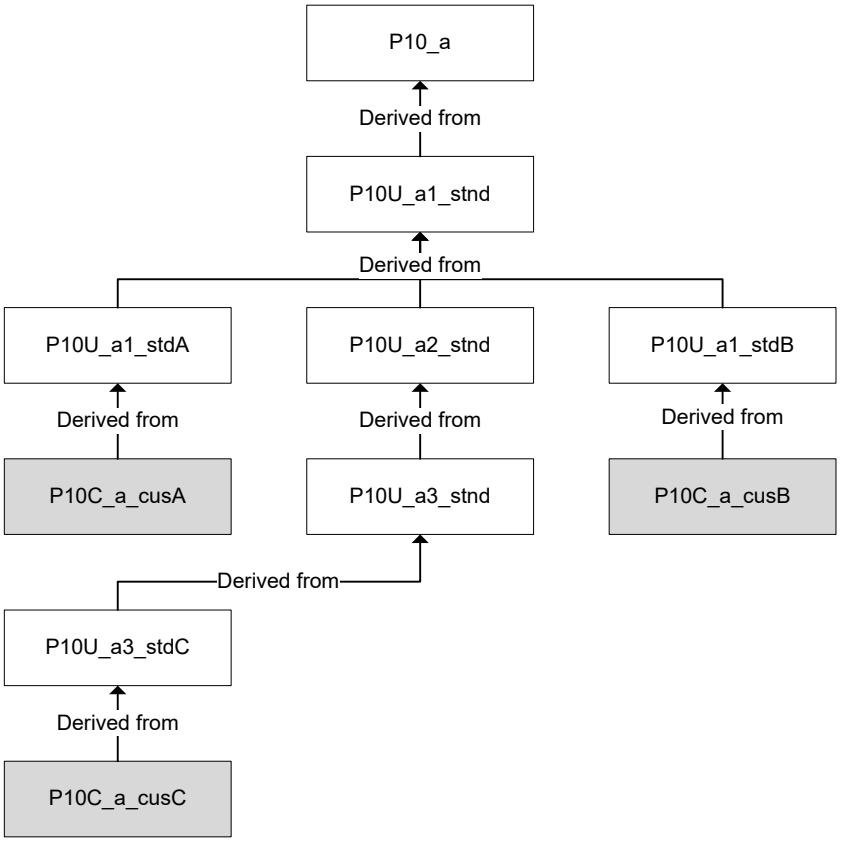
In most cases, you use a single environment to develop and to maintain multiple customizations. You build every customization on a specific patch. In addition, you must periodically upgrade existing

customizations to a newer patch. Therefore, you must have multiple Feature Packs of the standard parent product that is installed in separate VRCs. The best way to achieve this is this method:

Install patch 1 of the standard parent product in update VRC P10U_a1_stnd. When patch 2 is available, install the patch in a new, separate update VRC P10U_a2_stnd, which is derived from P10U_a1_stnd. Patch 2 has patch 1 as a prerequisite. You cannot install patch 2 in the new update VRC P10U_a2_stnd because patch 1 is not installed in this new update VRC. Therefore, run the Copy Solution Registry to Derived Update VRC (ttpmc2290m000) session and copy the registry of update VRC P10_a1_stnd into P10_a2_stnd. After you perform this step, you can install patch 2 directly in the P10U_a2_stnd update VRC. You can then repeat this same process for subsequent patches. The P10U_a<n>_stnd VRCs must only contain net patches. No additional solutions must be installed in these VRCs that do not belong to the relevant patch.

At this point, you have a VRC tree in which the net patches are installed in separate VRCs. In general, customizations are developed based on a net patch. Sometimes customizations are not derived from a net patch. Or a customer wants to install parent standard product solutions that have components which overlap with the customization. Therefore, the option to derive the customization VRC from any possible maintenance stage of the parent standard product must be available. The P10U_a<n>_std update VRCs are introduced to achieve this. For every customization an intermediate VRC layer is introduced in which standard solutions can be installed that do not belong to the underlying patch. You can use the Copy Solution Registry to Derived Update VRC (ttpmc2290m000) session to populate the PMC registry for the P10U_a<n>_std update VRCs.

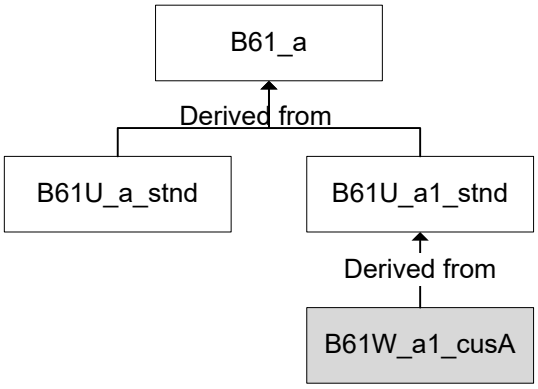
The customization VRCs P10C_a_cusA, P10C_a_cusB, and P10C_a_cusC represent various customizations that are derived from various patches. Suppose customization P10C_a_cusA is developed based on the net patch 1. During development of the customization, the P10U_a1_stdA VRC is still empty. After the customization is released and the customer installs the customization, the customer might want to install solutions of the standard parent product that also require an update of the customization. In this case, the customization vendor can install the solutions of the parent standard product in the P10U_a1_stdA VRC. Then port the solution to the P10C_a_cusA and provide a corresponding solution for the customization VRC. Installation of solutions in the P10U_a1_stdA VRC only affects the P10C_a_cusA customization and does not affect the other customizations in the environment.



Feature Packs and Customizations

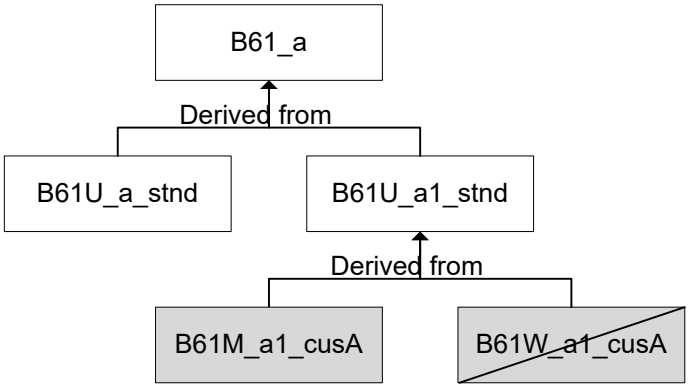
The system setup is almost comparable to the setup that is described for localizations and extensions in "Feature Pack development".. The main difference lies in the fact that Feature Packs for the parent standard product are now installed in an update VRC with PMC. Another differentiator is the fact that Feature Packs are cumulative, meaning that Feature Packs also contain all preceding Feature Packs.

When upgrading a customization to a higher Feature Pack level, having a VRC that only contains the net changes of a Feature Pack of the standard parent product can be handy. Suppose you must create a customization on top of Feature Pack 1 for a parent standard product. Feature Pack 1 is installed in the B61U_a1_stnd update VRC and a new VRC B61W_a1_cusA is developed on top of the Feature Pack.



The customization is built on top of the standard Feature Pack and is delivered as a Feature Pack in PMC. No master media must be created for the customization. When Feature Pack 0 for the customization is available, the Feature Pack is installed in the maintenance VRC B61M_a1_cusA.

Note: In theory, you can also skip the installation of the FP in the B61M_a1_cus1 VRC and start maintenance immediately in the B61D_a1_cus1 VRC. Best practice is to install and maintain the official build that customers also receive.



Solutions for Feature Pack 1 of the parent standard product are installed in the B61U_a1_stnd VRC. When you scan solutions for the parent standard product, you can also print a customization report. The customization report provides an overview of which components must be updated in the B61M_a1_cusA VRC. When you install the standard solution, you can merge the changes to the B61M_a1_cusA VRC and create a corresponding PMC solution for the customization.

The approach for subsequent Feature Packs:

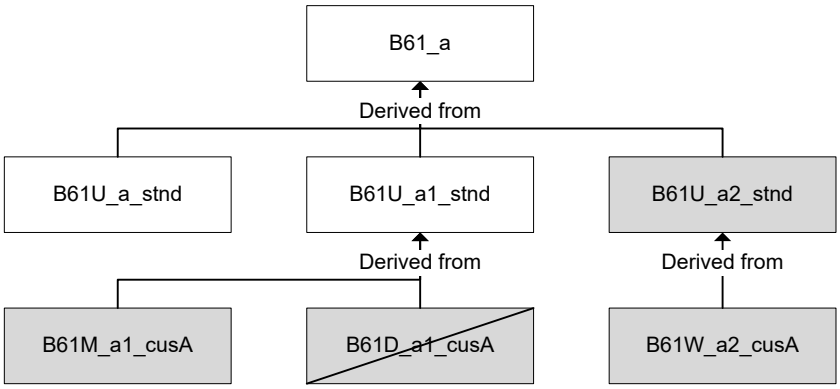
Install Feature Pack 2 for the parent standard product in the B61U_a2_stnd VRC, that is derived from B61_a VRC. Copy the contents of the B61M_a1_cusA VRC to the B61W_a2_cusA VRC. All maintenance changes you create on top of Feature Pack 0 of the customization until the copy action become available in the B61W_a2_cusA VRC. After the copy, you must port all maintenance changes in the B61M_a1_cusA VRC to the B61W_a2_cusA VRC. Feature Pack 2 for the standard product is cumulative.

It means that all changes that are implemented in Feature Pack 1 are also present in Feature Pack 2. Create an overview of all components that were changed in Feature Pack 2 as compared to Feature Pack 1. These components must be present in the customization. You can use the Compare Package VRC's (ttadv6450m000) session to achieve this.

This table shows the values that you must specify:

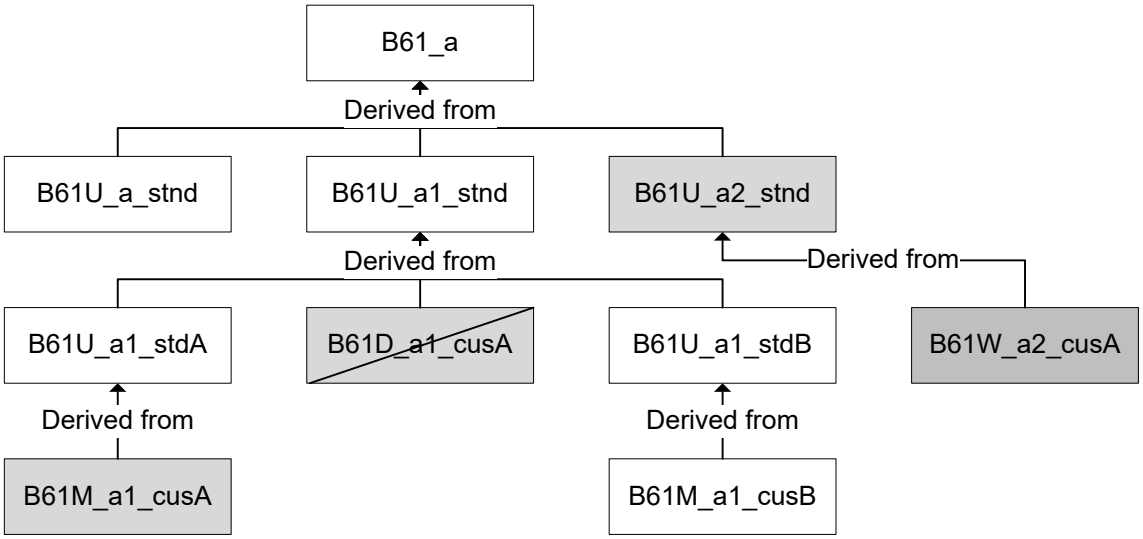
VRC 1	B61U_a1_stnd
VRC 2	B61U_a2_stnd
Where Software Components present in VRC	B61W_a2_cusA

With these settings you can create a net overview of customized components to be compatible and updated with the Feature Pack 2 of the parent standard product. You can then merge all changes that are implemented in Feature Pack 2 of the standard product from the B61U_2_stnd VRC into the B61W_a2_cusA VRC. You can then build Feature Pack 1 for the customization, based on the contents of the B61D_a2_cusA VRC. See this diagram:



You can repeat this procedure for an unlimited number of Feature Packs. In addition, you can develop and maintain multiple customizations in parallel on various levels of the standard parent product.

If multiple customization VRCs are derived from, for example, the B61U_a1_stnd VRC. Then for every customization, an additional update VRC layer B61U_a1_std is required between the customization VRC and the B61U_a2_stnd VRC. You can use the additional update VRC layer to install additional solutions of the parent product, which you must port to the customization. You must use the Copy Solution Registry to Derived Update VRC (ttpmc2290m000) session to populate the PMC registry for the B61U_a1_std update VRCs. Installation of additional solutions in the B61U_a1_stdA update VRC only impacts the B61M_a_cusA VRC and does not influence the B61M_a1_cusB VRC.



Dependencies for customizations

Individual solutions, patches or Feature Packs for customizations must have dependencies to corresponding updates of the parent standard product. Not having these types of dependencies can result in runtime compatibility problems. For example, if only a solution for the standard parent product is installed, and the installation of an update for the customizations was also required.

To establish these dependencies, define co-requisite dependencies between the updates of the parent product and the corresponding updates for the customized product. The co-requisite dependencies are generated automatically if the customization uses the same solution numbers that were also used in the parent standard product. This method only works if the customization base VRC, and the standard product base VRC, are part of the same base VRC combination at the distributor side. At the recipient side, the update VRCs for the parent standard product and the customization must also be included in a VRC combination.

The co-requisite relationship is only in one direction, from the customization to the standard parent product. You cannot install a solution for the customization without simultaneously installing the corresponding standard solution for the parent product. When you have not scanned the customized solution, you can still install a solution for the standard parent product without simultaneously installing the corresponding solution of the customization. This can result in run time compatibility problems. The customization report solves this problem. When you check standard solutions in a customized environment, you can optionally print a customization report. This report signals customized components and informs the recipient that, together with the standard solution, an update for the customization must also be installed.

Other

These sections are also applicable to the development and maintenance of customizations:

- "Rules to upgrade derived products"

- "Building Feature Packs and the Infor Installation Wizard"

PMC distributor session summary

This table shows the PMC distributor sessions:

Title	Description
Parameters (ttpmc0100s000)	Maintain the parameters for the PMC module.
Base VRC's (ttpmc0110m000)	Maintain the relation between *base VRCs* and *export VRCs*.
Base VRC Combinations (ttpmc0111m000)	Display the base VRC combinations.
Base VRC Combinations (ttpmc0111m100)	Maintain the base VRC combinations.
Base VRCs by Base VRC Combination (ttpmc0112m000)	Link the base VRCs to a base VRC combination.
Solutions (ttpmc1100m000)	Maintain solutions.
Collections (ttpmc1101s000)	Maintain *collections*.
Patch Header (ttpmc1106m000)	Maintain patch headers.
Packages by Patch Header (ttpmc1107m000)	Link packages to a patch header.
Defects by Solution (ttpmc1110m000)	Specify which defects are solved by a solution.
Component by Solution (ttpmc1120s000)	Link a component to a solution.
Sessions by Component (ttpmc1125m000)	Display the sessions that use a component that is linked to a solution.
Dependencies (ttpmc1140m000)	Define the dependencies between solutions or *patches*.
Solutions by Collection (ttpmc1151m000)	Add solutions to a *collection*.
Export Solution/Patch (ttpmc1200s000)	Create the export dump for a solution or a *patch*.
Export Solution Multi-level (ttpmc1202s000)	Create a multilevel export dump for a released solution.
Export Collection (ttpmc1210s000)	Export a *collection*, it means that a dump of the collection is made.
Connect Components to PMC Solution (ttpmc1221m000)	Connect software components in the *export VRC* to a PMC solution.
Generate Dependencies (ttpmc1240s000)	Generate the dependency relations between solutions which must be installed together at a customer system.

Title	Description
Base VRC not found	Generate a *patch*.
Generate Collection (ttpmc1251s000)	Add a range of solutions to a *collection*.
Generate Obsolete Solutions (ttpmc1252s000)	Generate obsolete solutions.
Generate Patch for New Base VRC (ttpmc1253s000)	Generate patch for New Base VRC.
Select "In Progress" Solution (ttpmc1299m000)	Link a component to an **In progress** solution.
Print Solution/Patch (ttpmc1400m000)	Print solutions and *patches*.
Validate Patch (ttpmc1401s000)	Validate a *patch*.
Validate Collection (ttpmc1402s000)	Validate a *collection*.
Print Collection (ttpmc1403m000)	Print the contents of a *collection*.
Validate Solution (ttpmc1404s000)	Validate a *solution*.
Print Components by Solution (ttpmc1420m000)	Print components that are connected to a solution.
Print Dependencies (Multi-level) (ttpmc1440m000)	Print the dependencies.
Print Maintenance History (ttpmc1460m000)	Print the history of changes for a solution or *patch*.
Patches (ttpmc1501m000)	Maintain *patches*.
Patches (ttpmc1501m100)	Maintain patches
Collections (ttpmc1503m000)	Maintain *collections*.
Components by Solution (ttpmc1520m000)	Connect the modified components to a solution.
Components by solution (ttpmc1521m000)	View the components that are linked to a solution.
Dependencies (ttpmc1541m000)	View the dependencies between solutions or *patches*.
Solutions by Collection (ttpmc1551m000)	View the solutions that are part of a *collection*.
Maintenance History (ttpmc1560m000)	Display the status changes of a solution or *patch* for a *base VRC*.
General Maintenance History (ttpmc1561m000)	Display the status changes of a all solutions and *patches*.
Specify Another Solution (ttpmc1821s000)	Specify another solution.
Post-Release Dependencies (ttpmc1141m000)	Add dependencies to a solution that was already released.

PMC solutions check

After a solution was released, it can happen that an error in this solution is discovered.

Therefore, the PMC distributor must create a new solution that has a post-requisite dependency to the incorrect solution. It can also happen that the PMC distributor has forgotten to add a post-requisite or prerequisite dependency of the solution. If the solution is already released, the distributor must add this dependency afterwards.

To provide this information to the customer, the distributor creates a `PMCDepend` file in the distributor part of PMC. This file is placed on the FTP support site.

The `PMCDepend` file contains this information for the PMC Base VRC:

- All post-requisite dependencies.
- All forgotten post-requisites and pre-requisites. This information is based on the dependencies that the distributor has specified in the **Post-Release Dependencies (ttpmc1141m000)** session.

The `PMCDepend` file has this format:

```
V/R/C/solution/V/R/C/requisite/enum constant for pre or post requisite/enum constant for active or obsolete
```

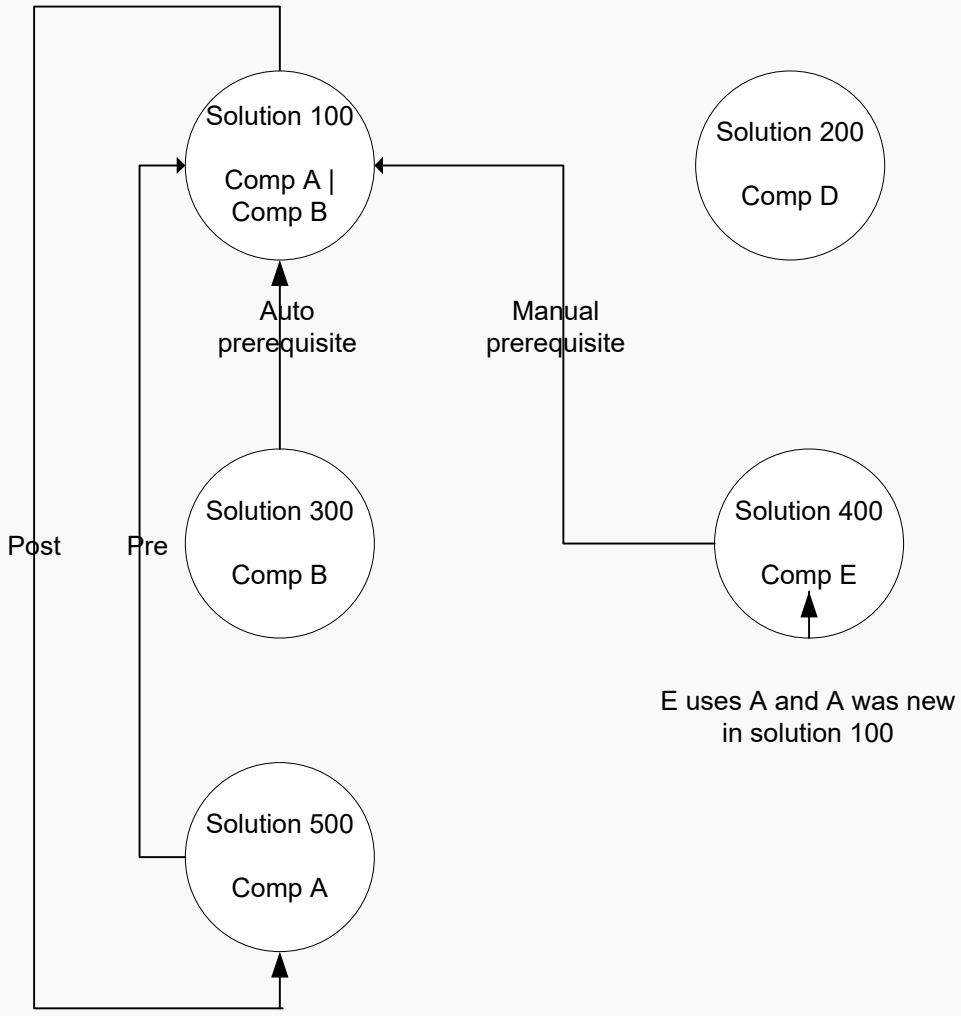
This code shows examples for Base VRC B61_a8, which is seen as B61_a8_stnd:

- `B61/a8//100/B61/a8//500/2/5`- Add solution 500 for solution 100 as post requisite.
- `B61/a8//400/B61/a8//500/1/5`- Add solution 500 for solution 400 as pre requisite.
- `B61/a8/stnd/300/B61/a8/stnd/100/1/10`- Remove solution 100 for solution 300 as pre requisite.

At the recipient side, the `PMCDepend` file is downloaded and scanned. For each solution of the current Base VRC, the dependency information of the `PMCDepend` file is stored in the Dependencies After Publish (ttpmc141) table. The imported dependency records exist forever and cannot be deleted. To undo such a dependency, the recipient must change the **Active** field of the corresponding row from **`Active`** to **`Obsolete`**. At the recipient side of PMC, during the “Check To Install”, the relevant dependencies are copied to the regular dependencies table (ttpmc140). Missing dependencies can automatically be downloaded. If the download fails, an error message is displayed in the report.

Example

This diagram shows an example of a solution structure:



Solution 500 is added after solution 100 was published. Customers who install solution 100 are forced to also install solution 500. After the `PMCDepend` file was scanned, also a dependency exists from solution 400 to solution 100. Therefore, if solution 100 is removed, solution 400 is also removed.

Distributor actions

The distributor must create and publish a `PMCDepend` file in these situations:

- A new solution, that has a post-requisite dependency to an already released solution, was added.
- A dependency was added afterwards to an already released solution.

To create and publish the `PMCDepend` file:

1 Start the **Solutions (ttpmc1100m000)** session.

2 Select the solution for which you want to create a `PMCDepend` file. You can only create a `PMCDepend` file for "non-obsolete" solutions with the **`Released`** status.
3 On the *appropriate* menu, select **Post-Release Dependencies**.
4 Specify the post-release dependencies. See the session help of the **Post-Release Dependencies (ttpmc1141m000)** session.
5 On the *appropriate* menu, select **Publish**.

Recipient actions

After the distributor has published a new `PMCDepend` file, the recipient must perform these actions:

1 Complete one of these steps:
   - If the latest `PMCDepend` file is already downloaded, scan the file. Use the **Scan Solution/Patch File (ttpmc2200s000)** session.
   - If the latest `PMCDepend` file is not downloaded yet, download and scan the file. Select the **Download Post Release Dependencies** check box in one of these sessions:
     - **Download, Scan and Connect Solution (Multi-Level) (ttpmc2210s000)**
     - **Check Solution/patch to Install (ttpmc2203s000)**
     - **Check and Install Solutions (ttpmc2204s000)**

2 Perform a dependency check to find out which solutions are missing:
   a Start the **PMC Dependencies Check (ttpmc2441m000)** session.
   b Specify the update VRC and click **Check**. A report is displayed. The report shows the solutions that are missing and indicates the required actions. See the session help.
3 Use the regular PMC procedure to download, scan, and install the missing solutions.

Chapter 7: User management

Here is explained how to create LN users and to authorize these users to use LN.

You can define the authorizations for LN in roles and templates that are linked to the LN users. The role and template concept provides you with a user-friendly method to quickly add new users or to update user authorizations.

Centralized user management through IFS

LN supports centralized user management. If user management is centralized, Infor Federation Services (IFS) is appointed as the System of Record (SOR) for security users.

In standard cloud installations, in which multiple customers are hosted simultaneously, this is the prescribed configuration. In other installations, this is the recommended configuration.

The security user integration between IFS and LN is role-based: in IFS, you must assign one of three specific, predefined standard roles to a security user. This makes the user an LN user, and grants the user access to the LN application through Infor Ming.le.

These are the standard roles that you can assign to a security user in IFS:

- Infor-SystemAdministrator - This role is used for overall administrators and makes the user a super user in LN.
- LN-Administrator - This role is used for LN administrators and makes the user a super user in LN.
- LN-User - This role makes the user a normal user in LN. To enable the user to actually use the LN application, you must assign additional LN-specific roles.

For details about centralized user management through IFS, see the *Infor LN Integration Guide for IFS*.

User management overview

You can use LN 's user management to define the user data and authorizations. The user's authorizations depend on their role in a company.

To work with LN, a user must meet these requirements:

- An LN user logon with a password, and the correct authorizations.
- A system logon for the operating system on which LN is installed, such as UNIX, or Windows. For information about user accounts on your operating system, see the appropriate Installation Manual.

LN user password

An LN user must be linked to an Operating System user.

When logging on to the LN application server with the logon procedure of Web UI or Worktop you must supply a password. The password that you must supply is not a special LN user password, but it is the Operating System password.

One of the System Administrator tasks is to define the password policy for the organization.

The password policy includes:

- Restricted passwords. For example you are enforced to supply a password of at least 6 characters and at least one numeric or special character.
- Password history, for example you cannot reuse recently used passwords.
- *Password aging*, for example you are enforced to change your password every two months.

To change your password

A System Administrator can give you a new password. You can change your own password if the System Administrator has given permission to change the password.

To change your own password, you must run the Change Password (ttstpchangepw) session. This session prompts for the current password and then prompts for the new password twice.

Note:

The session checks whether your user account is used to log on to a remote system. If so, the corresponding password in the Remote User Data (ttaad201) table is updated automatically.

Only run this session on a Master Application Server (MAS). If you run this session on an application server (AS), only the password in the remote user data on the AS is changed. The password on the MAS is not updated. Therefore, you cannot log on to the MAS anymore.

If the `passwd` command in the UNIX operating system of the LN server expects input parameters other than `old password- new password- new password`, an error message is displayed. For details, refer to the online help of the **Change Password (ttstpchangepw)** session.

To change your password through the Operating System

Alternatively, you can change your password through the Operating System of your LN server:

- In a Windows environment you can change your password in the User Accounts section of the Control Panel.
- In a UNIX environment you can change your password on the Command Line. You can, for example, use the `passwd` command.

Note: If your user account is used to log on to a remote system, you must update the corresponding password through the **Remote User Data (ttaad2501m000)** session.

Password Aging

Password Aging is supported for Web UI and Worktop in a Windows Environment.

In a UNIX environment Password Aging for Web UI and Worktop is only supported if *Pluggable Authentication Modules* are installed and enabled.

If your password is nearly expired, you are prompted to change your password.

If you decide to change your password, a dialog box, where you must specify your old and new password, is displayed. After your password is changed, a confirmation message is displayed.

Note: The session checks whether your user account is used to log on to a remote system. If so, the corresponding password in the **Remote User Data (ttaad201)** table is updated automatically.

LN user types

LN users are divided into two groups:

- Normal users: These users have restricted authorizations to start sessions, access table fields, work with data that pertains to specific company numbers, and so on.
- Super users: These users have unrestricted authorizations. A system administrator is a good example of a super user.

With the correct authorizations, developers can customize LN in a development environment. A development environment in LN consists of a company, a package combination, and multiple package VRCs.

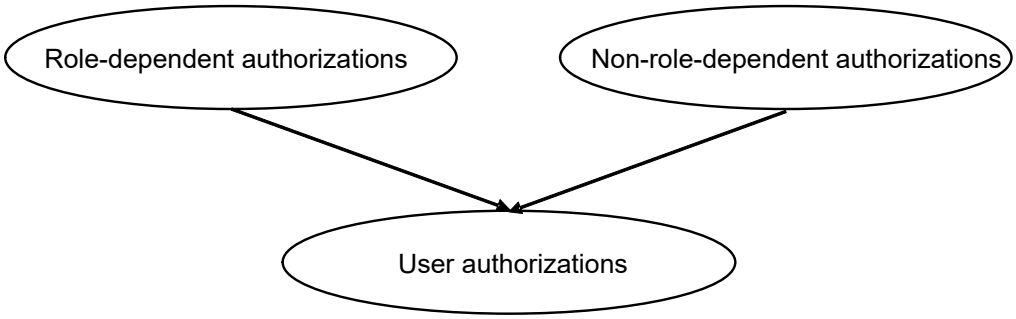
LN user authorizations

Initially, Normal users cannot use LN. Therefore, you must define some authorizations for the various LN software components.

The authorizations of an employee in a company are related to the employee's functionality or role in that organization. Some employees have more authorizations than others. Likewise, LN 's Authorization Management System (AMS) also uses a role concept to define the authorizations of LN users.

In addition to these role-dependent authorizations, you can define some additional dependent authorizations that are not defined by the employee's role. For example, the development parameters, device preferences, and other authorizations. These non-role-dependent authorizations are defined in templates, which you can connect to the user profile.

This diagram shows a schematic overview of how the user authorizations are split up into role-related authorizations and non-role-dependent authorizations:



Note: At minimum, users must have some sort of session authorization, table authorization, and library authorization to use LN. Some default roles are already automatically generated to ensure normal users have sufficient authorization to logon.

Session Authorization (Enterprise Modeler, AMS and Enhanced AMS)

Use these methods for defining employee roles:

- Infor LN Enterprise Modeler (EM)
- Authorization Management System (AMS)
- Enhanced AMS, that fully integrates Enterprise Modeler with AMS

Enterprise Modeler

In Enterprise Modeler, an organization can be modeled, including Enterprise Modeler Business processes. Enterprise Modeler Business processes contain activities that should be executed, such as paying an invoice, or raising a sales order. These activities can be LN sessions.

An LN user is linked one-to-one to a Enterprise Modeler employee. This employee has one or more Enterprise Modeler Roles. These Enterprise Modeler Roles are linked to one or more Enterprise Modeler Business processes. This way the access to the LN sessions is controlled from user login to LN session.

At runtime, an employee has a specific Enterprise Modeler menu, the Process browser. When an employee runs LN sessions from the Process browser, the modeled Enterprise Modeler authorizations are used to authorize all sessions in the menu. The Enterprise Modeler authorizations are deduced from the modeling information and are not stored in any static authorization tables.

If you start sessions from the Process browser, LN ignores any roles with session authorizations that are defined in AMS.

The Enterprise Modeler roles and authorizations are not applicable if you start sessions from the Menu browser. They are only applicable if you use the Process browser.

AMS

In AMS, you can define roles with authorizations for sessions, tables, and table fields. The roles can be linked to Normal users. By default, Normal users do not have any authorization. Super users have Full authorization for all sessions and all tables.

You can define different authorization levels, such as Full, Read Only, or Not authorized.

The roles that are linked to an LN user account determine the authorizations. For example; what is allowed to execute in the LN Menu browser and what is the authorization level.

The AMS authorizations are applicable if you start sessions through the Run Program command.

Enhanced AMS

This method is available to link Enterprise Modeler and AMS. You must explicitly enable Enhanced AMS in the AMS parameters, previously known as SSO Parameters. If LN is installed as a new installation, this is the default. If Enhanced AMS is activated, the **Tools > User Management > Authorization Management System** menu contains the corresponding new sessions. The other sessions are removed.

If you use the Enterprise Modeler modeler, we recommend that you use the Enhanced AMS with Enterprise Modeler.

To enable Enhanced AMS with Enterprise Modeler:

1. Select the **Support Export of EM Roles to AMS** check box in the **AMS Parameters (ttams0100m000)** session. Now, Enterprise Modeler is used to model the organization and business activities, and AMS is used to control authorizations.
2. Run the **Aggregate Modeled Authorizations for AMS (tgbrg9298m100)** session. This session gathers the data from the selected Enterprise Modeler version and project model, and exports the data to AMS.

If a session appears multiple times in the selected Enterprise Modeler version and project model, the widest authorization is used when aggregating the data. This table shows an example:

Enterprise Modeler session occurrence 1	Enterprise Modeler session occurrence 2	After aggregate
No authorization	Display	Display
Full	Display	Full

If you use Enhanced AMS, Enterprise Modeler subapplications are handled differently than if you use only Enterprise Modeler. See this list:

- If you use only Enterprise Modeler, a subapplication that is not specifically modeled has the same authorization level as the main session it is part of. That is, the authorization level is inherited. Not specifically modeled means not given an authorization level in the Enterprise Modeler module.
- If you use Enhanced AMS, a subapplication that is not specifically modeled is not included in the AMS authorizations.

When the menu for the user is created using the Process browser, the most restrictive authorization of Enterprise Modeler and AMS is used. This table shows an example:

Enterprise Modeler	AMS	Runtime
Display	Full	Display
Full	Display	Display

Enterprise Modeler	AMS	Runtime
NA or blank	Full	NA
Full	NA or blank	NA

Suppose, a subapplication is not specifically modeled in Enterprise Modeler, and no authorization is granted in AMS. In an environment that uses Enhanced AMS, the user does not have any authorization to run this subapplication.

Using Enhanced AMS might look more complicated than using Enterprise Modeler authorizations, but it gives huge advantages in reporting and controlling the authorizations. The authorized sessions and subapplications are all clearly specified if they are required. This results in much less unnecessary authorization settings.

Enhanced AMS is required for the integration with Infor Risk & Compliance Authorization Insight (IRC). The data that is shared with IRC only has AMS as its source.

The AMS role modeler can still change or overrule a specific AMS role. To activate Enhanced AMS, you must perform a Convert to Runtime of all roles and all users.

You can actualize the Enterprise Modeler data and convert these to Runtime data in one go, without any action or authorization required in the AMS environment. To achieve this, use a parameter setting in the **AMS Parameters (ttams0100m000)** session.

Printing session authorizations

Customers require a clear overview of the authorizations of a certain employee for the LN applications. This is in connection with the Sarbanes - Oxley Ac. Officially titled the Public Company Accounting Reform and Investor Protection Act of 2002.

These sessions are available to print the session authorizations:

- Enterprise Modeler

  Print Enterprise Modeler session authorizations (tgbrg8441m000)
- AMS

  Print Session Authorizations by User (ttams3400m000)
- Enhanced AMS

  No specific Print session exists. The preferred method is to use **Authorization Workbench (ttams4300m000)** session to view the AMS roles per user, role or session.

Role-dependent authorizations

From a user perspective, a *role* represents a functionality in a company.

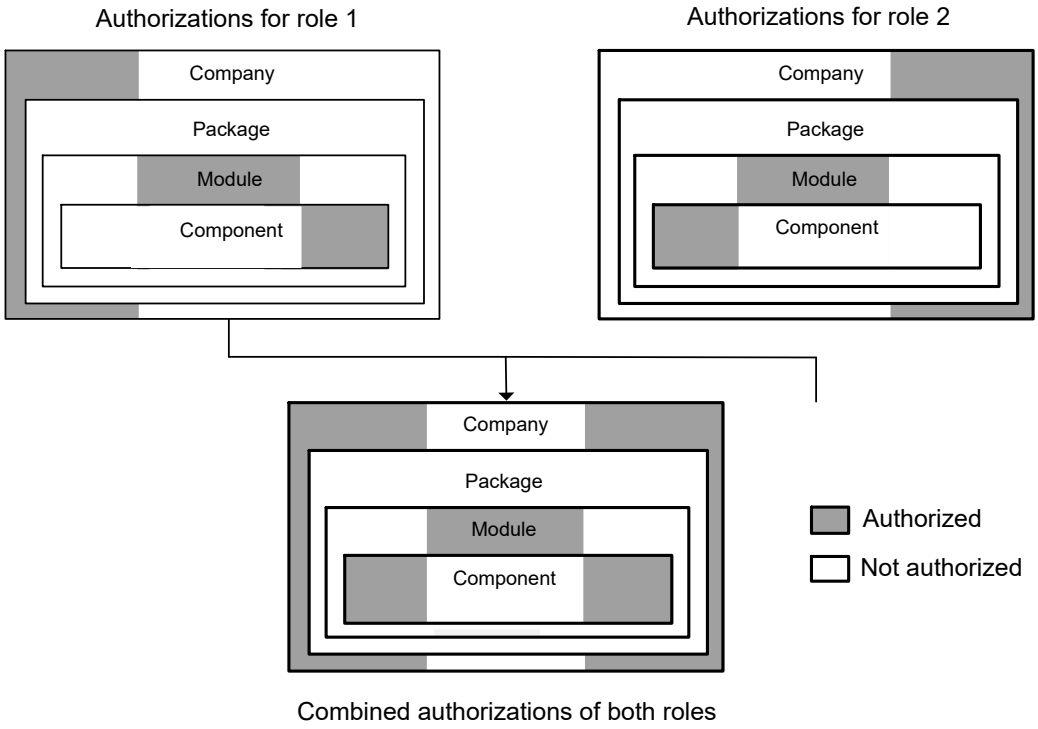
In LN 's Authorization Management System (AMS), a role represents a set of authorizations for a functionality in a company. User authorizations that are defined by role instead of by user significantly reduce the redundant data. The authorizations for normal users are, therefore, defined in roles to which

the authorizations can be linked. The role concept provides you with a user-friendly method to add new users or to update user authorizations.

Because an employee can have more than one functionality in a company, you can assign the user to more than one role. A role can also contain more than one sub-role, which itself can also have sub-roles. All these roles and sub-roles form a role tree, which you can view with the role browser. The role browser shows the role tree in a graphical user interface.

Ultimately, the employee's role is a combination of all the authorizations defined in the user's roles and sub roles. Recursive role structures are not allowed. For example, a junior software engineer cannot have the authorizations of a senior software engineer in a sub role.

This diagram shows an example of the combined authorizations of two different roles:



Combined authorizations of both roles

- Enhanced AMS

  With Enhanced AMS there is no limit to the number of roles that can be assigned to a user. A user can have multiple roles assigned which removes the necessity to support the role hierarchy.

For example, a department manager has more responsibilities than the employees in the department and therefore has more database authorizations. Consequently, the manager has two roles:

- The role of the employee with the appropriate restricted authorizations
- The manager's role with additional authorizations, which are only relevant for the manager

The restrictions on the table authorizations of the two roles are combined for the department manager. If the table authorizations are restricted for one role but not for the other role. The department manager ultimately has permission to carry out the database actions.

Role-dependent authorization types in LN

In LN, you can define the role-dependent authorizations at these component levels:

- Session authorizations
- Table authorizations
- Library authorizations

If the role-dependent authorizations change, conversion indicators are automatically set. Changes to the session authorizations, table authorizations, and the library authorizations are only converted to the run-time data dictionary when the conversion indicator is set. This is to avoid unnecessary conversion of the authorization data to the run-time data dictionary.

Session authorizations

The session authorizations define which sessions the users can start and what the users can do with these sessions in LN. You can specify the session authorizations and on several levels for either a specific company or for all companies. For example, you can give the users authorizations for only specified sessions in a module or only the sessions in a specified package.

The session authorization with the highest priority (1) is stated at the most specific level. The lowest priority (8) is stated at the most global level. The session authorizations that you define for a specific company have a higher priority than those defined for all companies.

This table shows the session authorization priorities:

	One company	All companies
Session authorizations per session	1	2
Session authorizations per module	3	4
Session authorizations per package	5	6
Session authorizations per company	7	8

In LN 's AMS, you can define the session authorizations with these sessions:

- Session Authorizations by Company (ttams3133m000) , which defines the session authorizations at company level
- Session Authorizations by Package (ttams3130m000) , which defines the session authorizations at package level
- Session Authorizations by Module (ttams3131m000) , which defines the session authorizations at module level
- Session Authorizations by Session (ttams3132m000) , which defines the session authorizations at session level

Enhanced AMS (the 'classic sessions' ttams3xxx are replaced):

- Role Data overview (ttams4100m000), which defines the roles.
- Role Data details (ttams4600m000), which defines the different authorizations (session, table, table field, library and role assignments).

Table authorizations

The table authorizations define the actions the users can perform on specified database tables and the associated fields in the database table. You can specify the table authorizations for a specific company, or for all companies, and on several levels. You can give the users authorizations for specified tables in a module or only some table fields in a database table and so on.

The table authorizations that you define in LN 's AMS are applicable to the databases for which the user is authorized. You can define the databases for which the user must be authorized in the RDBMS Administration module. For more information about the LN 's table authorization, see the "RDBMS administration" section.

The table authorization with the highest priority (1) is stated at the most specific level. The lowest priority (14) are stated at the most global level.

The table authorizations that you define for a specific company have a higher priority than those defined for all companies.

This table shows the table authorization priorities:

	One company	All companies
Database table field data authorization	1	2
Database table field authorization	3	4
Database table authorization per table data	5	6
Database table authorization per table	7	8
Database table authorization per module	9	10
Database table authorization per package	11	12
Database table authorization per company	13	14

You can define the database-table authorizations and the database-table-field authorizations with these sessions in LN 's AMS:

- Table Authorizations by Company (ttams3144m000)
- Table Authorizations by Package (ttams3140m000)
- Table Authorizations by Module (ttams3141m000)
- Table Authorizations by Table (ttams3142m000)
- Table Data Authorizations (ttams3145m000)
- Table Field Authorizations (ttams3143m000)
- Table Field Data Authorizations (ttams3146m000)
- Enhanced AMS: Role Data (ttams4600m000)

Note:

Table field authorizations and Table Field Data Authorizations have no effect on reports. If a user has no authorization at all for a table field, the field is still printed.

The database table field authorizations (1 – 4) only relate to the fields on the form. They have no effect on the database table authorizations (5 – 14) and are handled by 4GL.

Library authorizations

LN uses the *Business Object Layer* (BOL) integration technology, and OLE, DDE, OCX, and ORB interfaces to integrate programs with the LN environment. These programs communicate with LN through the Dynamic Link Libraries (DLLs). The Library authorizations define whether the users who are linked to the role can access the functions that are defined in DLLs.

See To Model a Business Object in the Infor Enterprise Server Web Help.

You can specify the library authorizations at several levels. You can give the users authorizations only for specified libraries in a module or only the libraries in a specific package and so on.

The library authorization with the highest priority (1) is stated at the most specific level. The lowest priority (3) is stated at the most global level.

This table shows the library authorization priorities:

Library per library	1
Library per module	2
Library per package	3

You can define the library authorizations at the various levels with these sessions:

- Library Authorizations by Package (ttams3150m000)
- Library Authorizations by Module (ttams3151m000)
- Library Authorizations by Library (ttams3152m000)
- Enhanced AMS: Role Data (ttams4600m000)

Non-role-dependent authorizations

The user data that is not related to the user's role can also be grouped to reduce redundant data.

To reduce this data, you can use LN 's AMS to create templates, which contain the user's authorizations that do not depend on the employee's role. The templates can also contain additional parameters that are required for developers to customize software components in LN.

The templates contain data that a group of users share. The templates are linked to individual users. The LN templates offer a user-friendly method to add new data. Or to update data for a group of users by linking the users to the templates.

The templates in LN 's AMS can be divided into these categories:

- User data and development-related templates
- Text-related templates
- Device-related templates

User data and development-related templates

In these templates, you can enter the default settings and parameters for a group of ordinary LN users and for LN users who develop software components in LN.

The data a user needs to maintain or create software components is defined in these templates:

- User-data template
- Developer-parameters template
- Developer-authorizations template

User-data template

You can use the user-data template to define the common user data that by a group of LN users share. The template contains the required system data and some non-role-related authorizations. The user-data template is used as an example in “To create templates,” later in this chapter.

You must use the User Data Template (ttams1110m000) session to create the user-data template.

Developer-parameters template

You can also use the template parameters that application developers use. For example, you can choose to run an automatic compilation after you create or change menus or forms. You can also specify commands and options, which the user can employ in the development environment.

Application parameters are settings the application developer requires to develop software components. Parameters are available for:

- Automatic compilation to the run-time data dictionary after changes to forms or menus.
- Actions after the **Copy to Current Package VRC** option.
- The parameters that the editor can use to develop software.

For every LN user, the data in this template is dumped to LN ’s run-time data dictionary in the User Application Parameters (ttadv010) table.

You must use the Development Parameters Template (ttams1150m000) session to create the developer-parameters template

Developer-authorization template

You can use this template to define some of the authorizations for developers who must customize LN software components. This session is password-protected. You can only use this session if you have system administrator’s rights.

In this template, you can specify this information:

- The package VRC for which the developer must be authorized to customize and develop software components. This specific package VRC overrides the **Default Authorization for all Package VRCs** check box in the **User Data (ttaad2500m000)** session.
- The languages and modules of the specified package VRC for which the developers must be authorized to customize and develop software components.

When selecting the **Authorization for all Modules** and **Authorization for all Languages** check boxes, the users who are linked to the template are authorized to maintain and develop software components in all LN modules and in all languages. If the check boxes are cleared, you must specify the specific modules and languages for which the user must be authorized.

If you select the **Components of other Developer** check box in this template, the user is authorized to maintain the software components that are created by other users. This option is helpful, for example, for a senior application developer.

The data in this template is dumped to LN 's run-time data dictionary for each individual LN user to the Version Authorizations by User (ttadv041) table.

You must use the Developer Authorization Template (ttams1151m000) session to create the developer's authorization template.

Text-related templates

A normal user must have some basic authorizations to use, update, or read text. The required data and authorizations are defined in the text parameters.

These text parameters are defined in these templates:

- Default text groups template
- Default text groups by text field template
- Text group authorization template

You can specify text parameters for a specific company or for all companies. When selecting the **All Companies** check box, the users who are linked to the templates are authorized to edit text in all companies. To restrict the users to a specific company, you must define that specific company in the **Comp** field. The text parameters that are defined for a specific company take precedence over defaults that are defined for all companies.

Default text groups template

You can use this template to define default text groups in LN. If a text is written in a text field for which no default text group is defined, the default text group is used. You must use the Default Text Groups Template (ttams1121m000) session to create the default text groups template.

Default text groups by text field template

You can use this template to define default text groups for text fields in LN. If text is written in a text table field, the text is linked to the default text group of that field. You must use the Default Text Groups by Text Field Template (ttams1120m000) session to create the default text groups by text field template.

Text group authorization template

Use this template for each text group to define the use, update and read authorizations. A text group defines how text must be presented in a window by defining the text editor, default window, and dimensions of the window. You must use the Text Group Authorization Template (ttams1122m000) session to create the default text groups authorization template.

Device-related templates

You can define the device-related data in the device preference template.

Device preference template

You can use this template to group devices and define an order of preferences for these devices. For example, you can define a range of printers, and give the highest priority to the printer closest to the user. The data in this template is dumped for each individual LN user to the Device Preferences (ttaad306) table.

You must use the Device Preference Template (ttams1140m000) session to create the device preference template. If the template is linked to a user who is not authorized for all devices. The user is only authorized to use the devices that are defined in this template. You can authorize a user for all devices if you select the Authorization for all Devices check box in the **User Data Template (ttams1110m000)** session.

Role browser

You can view the roles and sub roles in the role browser, which shows a graphical representation of the roles in a role tree.

The role browser uses the display logic of the existing desktop browser to ensure a consistent interface across the various browsers in LN.

The role tree represents a role with all its sub roles, which can also contain sub-roles. You can define the authorizations for the sub-roles in the same fashion as for the roles.

In the role browser, you can double-click a role folder to view the sub-roles in the role. To start the Role Data (ttams2100m000) session, select a role or sub role, and on the **Options** menu, click **Role Data**.

You cannot define a recursive role structure. For example, a junior software engineer cannot have the authorizations of a senior software engineer in a sub-role. The Role Browser identifies and highlights recursive roles. You must delete the highlighted recursive roles.

Enhanced AMS

The Role browser is replaced by the **Role Data (ttams4100m000)** overview session and **Role Data (ttams4600m000)** details session. The notion of role tree is removed in Enhanced AMS, for example, there is no longer a hierarchy of roles.

User-related procedures

The User Management module contains the procedures that you must use to authorize employees to use LN.

With these procedures, you have these options:

- Create LN users.
- Create the remote user accounts for the LN users in a client/server environment.
- Change the *package combinations* for LN users
- Maintain the user's developer authorizations.

Procedure result

The user can start LN, use the Menu Browser, and start sessions. The user can also use LN in a client/server environment. Developers have the correct development authorizations. The LN user also has permissions on a database level that can be restricted with LN 's Authorization Management System (AMS).

Prerequisites

- The user must have a system account for the operating system on which LN is installed, such as UNIX or Windows. If the user does not have a user account on the operating system, refer to the appropriate Installation Manual for your specific operating system.
- On a DB2 installation, you must also add the user to the LN db group with the Windows User manager.
- If the licensing of LN is based on named users. The LN user accounts that are created with User Management must correspond with the user names in the named user list in the Solution License Manager (SLM).
- To create remote user accounts for LN users, the LN user must have a user account on the local system or workstation.
- The roles and templates to which you can link the normal users are already defined.

  If these components are not defined, see the Creating roles and authorizations on page 88 and Creating templates on page 89 sections.
- The development-environment data for the developers must be defined.

This section describes only the most important steps in the procedures. For detailed instructions on how to enter data in the described sessions, see the online help of the sessions.

Creating LN users

To work with LN, a user must have a user logon, a password, and the correct authorizations.

The authorizations depend on the user's role in a company. You can use the User Management module to define the user's user data and authorizations.

Upon completion of this procedure, the new users can log on. If you changed the user data for existing users, these users must log off and log on again to use the new settings.

This list shows the procedure steps and the corresponding sessions.

1. Define the basic user data - User Data (ttaad2500m000)
2. Convert the changes to the runtime data dictionary - Convert Changes to Runtime DD (ttams2200m000)
3. Create a database user - Convert Infor LN User to Database User (ttdba0915m000)

Creating remote user accounts

If the LN database runs on a remote system, these situations can occur:

- The database driver runs on the (local) application server and communicates with the remote database through the database client software. In this case, you do not have to create remote user files. This case is the most common situation.
- The database driver runs on the remote database server. In this case, you must create remote user accounts for the LN users. The LN users connect to the remote system through remote user accounts. You can create remote user accounts on the local system or from the remote system.

If you create the remote user file from another remote system. The Startup System field in the user template must contain the user's system name.

Ensure that the application manager or system administrator who creates the remote user account for the LN user's system. The LN user account is in fact a remote user account of the application manager or system administrator.

This list shows the procedure steps and the corresponding sessions.

1. Define the remote user data - Remote User Data (ttaad2501m000)
2. Convert the changes to the runtime data dictionary - Convert Changes to Runtime DD (ttams2200m000)
3. To apply the new settings, the relevant users must log off and log on again.

Changing the package combination for LN users

A *package combination* is a collection of various LN packages.

For example, the package combination B61a contains the packages LN Common, LN Financials, LN Manufacturing, and so on. Every package combination has a unique *package VRC*. The package combination links the LN users to a specific LN version.

If you change the package combination of a company, you must also change the user's package combination. You can only change the user's package combination into a package that matches the package combination of the user's default company.

The default company's package combination can only differ from the user's new package combination under these conditions:

- Both package combinations include the same packages.

- Various package VRCs are used in the package combinations.

The package VRCs can only differ in the package combinations in these instances:

- Both VRCs are derived from each other.
- Both package VRCs are derived from the same VRC and no changes in data definitions or domains have been made in the derived VRCs.

This list shows the procedure steps and the corresponding sessions.

1. Change the package combination for a specific LN user or a range of LN users - Change Package Combinations for Users (ttaad2200m000)
2. The relevant users must log off and log on again.

Changing the password for developer authorizations

The authorization data in the User Data (ttaad2500m000) session and the Developer Authorization Template (ttams1151m000) session are password protected.

To change the password for developer authorizations, start the Change password for Role and Developer authorization (ttadv0143m000) session.

Note: There is no default password.

Creating roles and authorizations

Initially, normal users do not have any authorizations for LN. Therefore, you must define the role-dependent authorizations for their function. You can use LN 's Authorization Management System (AMS) to create roles for normal users and the associated role-dependent authorizations and create templates.

The authorization management procedures create a user environment with clearly defined tasks and duties for the normal users in a company.

Prerequisites

These prerequisites are required to create roles and authorizations:

- The functions and associated tasks are clearly defined for the employees.
- The software components to which access is required for the functions are clearly defined.

The case study explains how to create roles and authorizations and helps you understand the functions and features of LN 's AMS. The case study also describes the role concept in the context of a real situation.

The case study describes authorization management at Global Enterprises, which has offices in The Netherlands and the United States. These offices are designated as the Holland Company and the USA Company. Enable employees of Global Enterprises to use the company's data dictionary and the databases of both offices.

For detailed instructions see the online help of the sessions

Procedure introduction

The authorization management procedure is split up into several smaller procedures. The system administrator can use them as stand-alone procedures to maintain the authorizations at the various levels. The additional roles, which define additional authorizations on top of the standard role, can also be created with this procedure.

Case study - Authorization management at Global Enterprises

The system administration department at Global Enterprises is responsible for management of the employees' authorizations. Authorization management involves the definition of the role-dependent authorizations for the employees in roles. To ensure that the user authorizations are maintainable. The system administrator defines one standard role with the most basic authorization, and additional roles for specific tasks.

The authorizations that are defined in the standard role are required by the users to use Global Enterprises' general data dictionary and the data of the offices in The Netherlands and the USA.

In the standard role, the system administrator defines the session authorizations and library authorizations at the various levels. In addition, the system administrator ensures that the users cannot change their own user data.

Procedure summary

This list shows the procedure steps and the corresponding sessions.

1. Take inventory of the roles in your organization
2. Define the authorizations per role - Role Data (ttams2100m000)

   Through the *appropriate* menu in this session, you can:

   - Define session authorizations.
   - Define database authorizations, table and table field authorizations.
   - Define library authorizations.
   - Optionally: define authorizations in sub-roles and link these sub-roles to the main role - Subroles by Role (ttams2101m000) session.
3. Convert the user file to the runtime data dictionary with the Convert Changes to Runtime DD (ttams2200m000) session.
4. The relevant users must log off and log on again.

Creating templates

A template contains common data for a group of users. In these templates, you can define the non-role-dependent authorizations for a group of normal users. You can also specify additional template parameters that a group of users share.

Templates provide the system administrator with an efficient method to define user data and additional parameters that are required for groups of LN users.

Prerequisites

To create templates:

- The functions in the company must be defined.
- The required system data, development parameters, text parameters, and so on must be defined for the employees.

Procedure Introduction (User Data template)

The User Data template is used as an example. The procedure is representative for most templates in LN 's Authorization Management System (AMS). You can convert changes to the template to the run-time data dictionary. On the *appropriate* menu in these sessions, click **Convert to Runtime DD** to start the Convert Changes to Runtime DD (ttams2200m000) session. For a detailed description of this session, see the session's online help.

When the template data has changed, conversion indicators are automatically set for all individual LN users who are linked to the templates. The user data is only converted to LN 's run-time data dictionary if the data in the templates has changed. The conversion is, therefore, only performed when necessary.

Procedure summary (User Data template)

This list shows the procedure steps and the corresponding sessions.

1. Take inventory of the user data that can be shared
2. Define the User Data template - User Data Template (ttams1110m000)

   For detailed instructions, see the online help of the session.
3. Convert the user template data to the runtime data dictionary - Convert Changes to Runtime DD (ttams2200m000)
4. The relevant users must log off and log on again.

Active Users

Running the **Active Users (ttaad2560m000)** session shows the list of Active users on the LN environment. The session monitors the logged in users and refreshes the list every five seconds.

Note: The **Active Users (ttaad2560m000)** session is only supported if the LN application server runs on Windows.

Running the session (Windows only)

The session shows these columns:

- Login Time

- User info
- Bshell PID
- Category (Background Process or User Process)
- Description of the startup process, and the command line of the bshell

Examples of Category and Description:

- When a user logs in using LN UI, the Category is User Process, with 'UI Process' as description.
- A bshell handling Job ABC is displayed as a Background Process, with 'Job Process [ABC]' as description.

You can manually stop the bshell of an Active user with the Terminate command. The BaanLogin Daemon stops the selected bshell. Note that when you are logged on as a **`Normal User`** you only can only stop your own processes. To terminate the processes of other users you must log on as a **`Super User`**.

Every bshell that is started creates a `bshell.<pid>` file in the `$BSE\proc` folder, containing the mentioned properties of the bshell. If a bshell process is stopped in an abnormal way the proc folder can become out of sync. The BaanLogin Daemon removes the remaining `bshell.<pid>` files in the `$BSE\proc` folder when the accompanying bshells do not exist.

Active Users BSE Service

The Active Users BSE Service compares every minute the files in `$BSE\proc` to the records in the Active Users session. The BSE Service ensures that Active Users tables in the database are in sync with the files in `$BSE\proc`. A cleanup process is started when there is a mismatch. and the database is brought in sync with the `$BSE\proc` folder again.

Electronic Signature

E-signatures are used to sign off documents for legal purposes or for strict internal control.

Examples are, a Bill of Material, a Shipment or a Payment batch.

For the documents you can specify several qualifications:

- In which situations is an E-signature required.
- Who must sign off, based on the underlying business case.

The documents data with the signature is stored in the E-signature tables in the LN Tools environment, separate from the active LN applications. Storing the data separately in the LN Tools environment secures the data. The data can never be changed, including the related data such as names and descriptions are included.

To meet legal obligations, the stored data with the E-signature can be audited by institutions such as the FAA or FDA, and other certifying institutions. Internal control or auditors can also run audits to meet internal procedures, depending on internal business cases.

The data in E-signatures can bypass the GDPR regulations, as the name of the person who signs is not anonymized. Any other name or description in the E-signature is also not anonymized or erased. The data for E-signatures can never be archived or removed.

Note: E-Signature is not supported in combination with OCM or Workflow.

Troubleshooting

This section contains information about issues that can occur in user management data.

Fall back menu

When the user data becomes non accessible and no start menu can be found a Fall back menu is displayed to the user. This menu shows the cause of the problem and the option to contact the system administrator.

When clicking an option in the Fall back menu, the **Companies (ttaad1100m000)** session starts.

The user receives a Default role and can log on to temporary proceed the work until the issues are solved by the system administrator. The user can use the **run program** menu to specify a session number to work with.

To solve this issue, the system administrator must check the user data file of this user. Specify a correct start menu and other information if required and run the convert to runtime process.

Chapter 8: Audit configuration management

You use LN audit features to fully or partially log changes that users make to the LN database tables when they use LN sessions.

You can view or print the history of modifications.

You can also use the history modifications to update external databases with the LN Exchange package or with the Business Object Layer (BOL/BOR) functionality

The audit functionality is centered on the concept of audit profiles. You define which tables and fields are audited and when, in the context of an audit profile. To bundle profiles in the same functional area, you can relate the profiles to audit categories. You can export and import audit profiles with various options to enable a quick configuration. The audit trail is stored in sequence files, which are generated for each combination of company and table.

Audit trail and audit host settings

The audit trail and audit host settings provide the required information to create audit trails.

Before you can create an audit trail, this information is required:

- The size of the trail files, sequence files, that are created. Define the trail file size in the Audit Trail File Sizes (ttaud3135m000) session. The maximum number of sequence files for each table/company combination is 999. You cannot delete the currently active file. Specify a file size that make it possible to delete old files, and keeps at least the currently active file. The trail file size must be large enough to store the audit trail of a large transaction, otherwise, the transaction is stopped.
- The path to the directory where the audit trail files are stored. Define the paths in the Audit Trail Paths (ttaud3136m000) session.
- The security settings for reading, maintaining, and deleting the sequence files. Define the security settings in the Audit Trail Security (ttaud3137m000) file.
- The audit host settings: Define the audit hosts in the Audit Hosts (ttaud3130m000) session. You can define audit hosts for each company. Defining audit hosts is optional. If no host is defined for a company, the local system is used as host. If a master application server is used, and one or more other application servers, ensure to define audit hosts. Otherwise, the transaction IDs in the audit trail might not be successive.

Note: To activate these settings, you must use the Create Runtime Audit Definitions (ttaud3200s000) session. Select the correct check boxes to convert the settings to run time. Only the security settings are effective immediately, and must not be converted to run time.

Audit configuration procedure

A short overview of the procedure steps to configure the audit settings, assuming that no audit settings are yet present.

The procedure steps and the corresponding sessions:

1. Define the audit categories - Audit Categories (ttaud3100m000)
2. Define the company groups - Company Groups (ttaud3140m000)
3. Define which companies are related to the company groups - Companies by Company Group (ttaud3145m000)
4. Define the audit profiles - Audit Profiles (ttaud3110m000)
5. Define the tables that must be audited - Audit Tables by Profile (ttaud3120m000)
6. Configure the audit settings for specific fields - Audit Fields by Table (ttaud3125m000)
7. Convert the audit settings to run time - Create Runtime Audit Definitions (ttaud3200s000)

Audit - Additional functions

The audit configuration management sessions provide several additional functions.

You can export and import profiles through these sessions:

- Export Audit Profiles (ttaud3201s000)
- Import Audit Profiles (ttaud3202s000)
- Import Audit Profile from Additional File (ttaud3203s000) : You can use this session for profiles that are delivered with the LN software.

You can analyze the audit profiles, and view where specific tables and fields are used, with these sessions:

- Where Used Audit Tables (ttaud3521m000)
- Where Used Audit Table Fields (ttaud3526m000)

To help you to migrate to from an earlier version of LN, you can use the Audit Configuration Migration (ttaud3204s000) session.

To maintain the generated sequence files, you can use these sessions:

- Display Audit Sequences (ttaad4560s000) : Use this session to display information about the sequence files.
- Print Range of Audit Files (ttaad4461m000) and Print Range of Audit Files (Multi Lines) (ttaad4463m000) : Use these sessions to print the content of sequence files.
- Transaction Notifications (ttaud1510m000) : Use this session to view detailed information about all transactions in a specific company and table.
- Check Audit Files Integrity (ttaad4460m000) : Use this session to check the integrity of the sequence files.
- Purge Audit Files (ttaad4261m000)

Audit - General remarks

The commands that cause a table transaction to be audited, are only the commands that affect the table data. That is, the Insert, Update, and Delete commands.

Several table level commands that affect all rows in a table are also audited, such as Create Table, Drop Table, and Clear Table.

The audit configuration uses a positive approach. It means that you can only define the tables and fields that must be audited. You cannot define the tables and fields that must not be audited. To compensate for this feature, you can load lists of all packages, modules, tables or fields through the *appropriate* menu of the concerned sessions.

Because you can use tables and fields in various profiles with conflicting settings. These rules determine which setting takes precedence over the conflicting setting:

- A higher level takes precedence over a lower level. When you define a profile A that audits all tables in a module and a profile B that only audits one table in this module. The result is that if you convert these profiles to run time, that all modules in the package that are concerned are audited.
- The setting **Always** takes precedence over **Changed**. For example, profile A defines that a field must only be audited when the profile is changed. Profile B defines that this field must always be audited. Then the field is always audited.

These two tables show the result of conflicting audit type settings:

Package	Module	Table	Audit Type
tt	adv	*	Always
tt	adv	200	Changed
tt	adv	*	Always

Package	Module	Table	Audit Type
tt	adv	*	Changed
tt	adv	200	Always
tt	adv	*	Changed
tt	adv	200	Always

If company groups are also added to a profile. The result of the convert to run time action on these profiles is determined by these company groups also.

This table shows that result of conflicting audit type settings:

Package	Module	Table	Company	Audit Type
tt	*	*	001, 002	Changed
tt	adv	*	002, 003	Always
tt	adv	200	002, 004	Changed

Package	Module	Table	Company	Audit Type
tt	*	*	001, 002	Changed
tt	adv	*	002, 003	Always
tt	adv	200	004	Changed

Several combinations of audit type and field specification are possible. This table shows the same information, but now by company, and only for table ttadv200:

Companies	Audit Type	Comment
001	Changed	2
002, 003	Always	4
004	Changed	
005	-	For other companies, table ttadv200 is not audited.

You can configure audit settings for tables in another package combination. You cannot zoom to these tables. You must enter these tables manually.

If you convert the audit configuration to run time, the result is stored in these four files, which you can find in the `$BSE\lib` directory:

- `audit_spec`
- `audit_cols`
- `audit_hosts`
- `auditdef6.2`

Audit - Specific issues

These sections describe the effect of several changes you can make in the audit settings.

The effect of a change depends on the specific situation. The situations that are described arise if changes in the audit settings are converted to run time and not all users did leave LN. Then, some users create audit trails based on the old configuration. Users that start afterwards create audit trails based on the new configuration.

Changes in the profiles

This table summarizes the effect of the changes for a particular table in a company:

Change	Consequence
Add a table.	Users that still use the old settings do not audit some transactions.

Change	Consequence
Remove a table.	Some transactions that must not be audited with the new settings are still audited by users that use the old settings.
Change the audit type for a table or field.	Some transactions are audited according to the old settings, and other transactions are audited according to the new settings.
Add or remove a field.	After you audit transactions in the table with the new settings, users that use the old settings can no longer perform transactions on this table. Therefore, users with the old settings can be forced to restart LN.

Note:

- If you switch field-specific auditing for a table on or off, and this results in a different number of fields to be audited. Then the affect of this change is the same as adding or removing a field.
- Not every change in the audit settings results in other settings at run time. If you convert the new settings to run time, the net result of the changes for the entire configuration might be zero.

Changes in the audit host

An important part of the audit trail is formed by the transactions IDs. To be useful, the transaction IDs must be successive. The transaction IDs are generated for each combination of table and company by the audit server, which runs on the audit host. If the audit host settings change, users with the new settings use another audit host. That host is another audit server than users with the old settings use. Therefore, the transaction IDs are not successive, and the audit trail is corrupted. The user does not notice this problem.

Changes in the directory of the sequence files

The directory where the sequence files are stored is changed. Users that use the old settings can still create new sequence files in the old directory. Therefore, the sequence numbers in the file names are no longer an indication for the sequence of the files. The user does not notice this problem.

Changes in the maximum file size of the sequence files

If the maximum file size of the sequence files is changed, and this change is converted to run time. The new file size is immediately effective. Both for users with the old settings, and for users with the new settings. If the file size was enlarged, the current file grows until this new size. If the file size was diminished, and the current file already exceeds this size. A new file is created the next time a transaction is logged.

Combination of changes

The changes that are described in the previous sections can also be combined. A combination of changes can have several advantages. A noticeable example is the combination of a change in the audit profile and a change in the directory where the sequence files are stored. This combination provides these advantages:

- The audit trails that are created with the new settings are not mixed with the audit trails created with the old settings.
- The users can continue their work.

Issues can still occur, beginning with the problem already described, that the sequence of the sequence files becomes unclear. A user can also start LN on the moment that the new audit settings are converted to run time. Then the user can use the new settings for the profiles, and the old settings for the directory of the sequence files.

Chapter 9: RDBMS administration

You can give LN users access to an RDBMS, and optimize the database table and index repository for an improved interaction between LN and the database.

Overview of RDBMS administration

Several *Relational Database Management Systems* (RDBMSs) are supported by LN for which you can authorize the *LN user*.

Before the LN user can access a database and can use the information that is stored in the RDBMS. You must give the LN user permission to access the RDBMS.

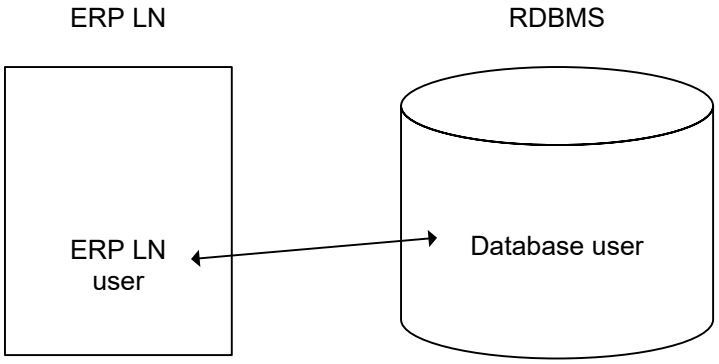
An RDBMS is a relational database or database management system that consists of tables made up of rows and columns. In a relational database, the rows of a table represent records and the columns represent fields with the particular attributes of a record. The records are collections of information about separate items. The attributes of the records are represented in the columns.

A database administrator (DBA) or system administrator manages an RDBMS. Tasks of a DBA are to determine the content, internal structure, and access strategy for an RDBMS. Defining security and integrity, and monitoring performance.

You can use the RDBMS Administration module for these tasks:

- Give LN users access to an RDBMS.
- Transfer the database user file and database group file to the RDBMS Administration module. You can use these tools if, during a migration to an updated LN environment, data is lost in the RDBMS tables.
- Optimize the table and index repository to optimize the interaction between LN and the database.

This diagram shows the link between the LN user and the *Database user*:



To make the data in the RDBMS tool manageable, these entities are used in LN:

- *Database group*
- Database user

Database groups

In LN, a database user must belong to a database group. A database group enables the DBA to assign a specific set of privileges to multiple users. All members of a group have the same privileges. Before you create a database user, the database group to which the user must be assigned must already exist.

A database group is already defined during the installation of the LN software.

Database group notes

A database base group is password protected.

The database group password is a default password and is not visible for the LN user.

An LN user has only access to the database through the LN application.

It is not allowed for an LN user to logon directly to the database.

Database user

A database user is a user who has access privileges to specific databases that are stored in the database server. The configuration information for the database user contains all necessary settings to log on to a database. These settings are automatically loaded when the LN user logs on. The group to which the user belongs determines the access privileges of the database user. Before you can add a database user, you must create a database group to which you can assign the user.

The table authorizations of an LN user are role-dependent. Therefore, you can restrict the LN user to those tables or table fields in the RDBMS that are applicable for the employee’s role in an organization. You can use LN ’s Authorization Management System (AMS) to define the database restrictions.

You can link a database user to one or more LN users (1 - many relationship).

Database user notes

A database user is password protected.

The database user password is a default password and is not visible for the LN user.

An LN user has only access to the database through the LN application.

It is not allowed for an LN user to logon directly to the database.

RDBMS administration procedures

You can use two procedures to authorize LN users for RDBMSs and to maintain the data in the RDBMS Administration module.

In LN 's RDBMS Administration you can find two main procedures:

- The recommended RDBMS Administration procedure to authorize LN users in a range or individually.
- The RDBMS Administration maintenance procedure. You can use this procedure to maintain the database users data, database groups data, and the link data. You can also use this procedure to change the database user's password and the database group's password.

These additional procedures are included in the RDBMS Administration module:

- Change the database group's password and the database user's password.
- Transfer the database group file and the database user file to the RDBMS Administration module.

When you completed the RDBMS Administration procedures, the LN users have access to an RDBMS.

Prerequisites

To authorize the LN user for an RDBMS, these prerequisites must be met:

- The employee user must have a system logon for the operating system on which LN is installed. If the user does not have a user account on the operating system, see the appropriate Installation Manual for your operating system.
- The employee must have an LN user account. In other words, the employee must be an LN user. If the employee does not have an LN account, refer to the procedure Creating an LN user.
- A database group must exist. The database group is usually created during the installation of LN.

  If the database group is not created successfully during the installation, see Recommended RDBMS procedure on page 101

Recommended RDBMS procedure

The recommended RDBMS Administration procedure is the best and fastest method to give LN users access to an RDBMS.

You can use this procedure for individual LN users or for an entire range of LN users. Giving a range of LN users access to an RDBMS is especially helpful during the installation of LN, or when a new an RDBMS is introduced.

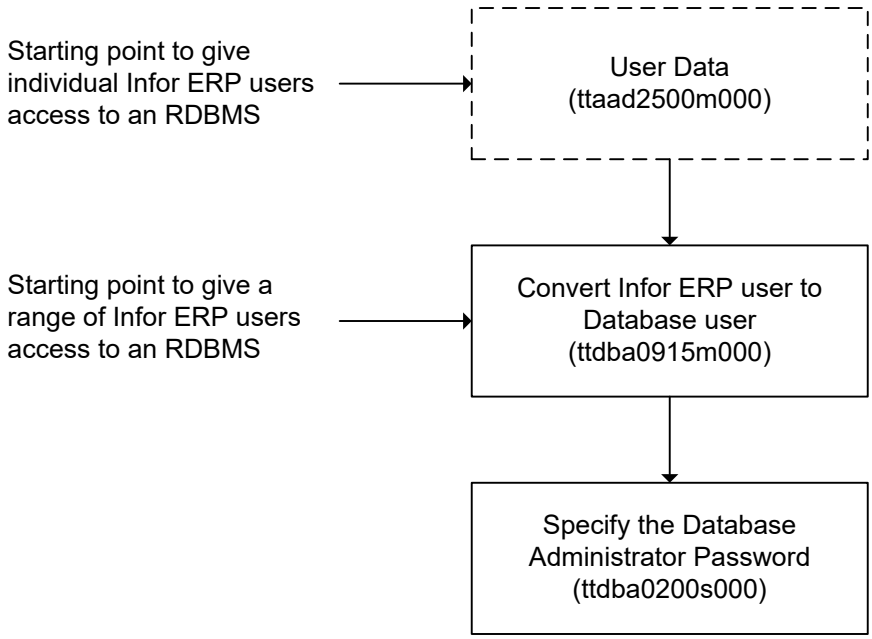
You can use the recommended RDBMS Administration procedure to convert LN users to database users. The link between the LN user and the database user is automatically created during the process.

To change default parameters, see RDBMS maintenance procedure on page 103.

Use the recommended RDBMS procedure to run these authorization options:

- Give individual LN users access to an RDBMS
- Give a range of LN users access to an RDBMS

This diagram shows the recommended RDBMS Administration procedure to convert LN users to database users:



How to specify data in the sessions, see the Infor Web Help.

To give individual LN users access to an RDBMS

You can use this procedure to give individual LN users access to a database.

You can restrict the LN user's database authorizations with LN 's Authorization management System (AMS).

Procedure summary

The procedure steps and the corresponding sessions.

- Select one LN user - User Data (ttaad2500m000) .
- Convert the selected user to a database user - Start the Convert Infor LN User to Database User (ttdba0915m000) session from the *appropriate* menu. You are prompted to specify the Database Administrator password. Use the Enter Database Administrator Password (ttdba0200s000) session.

To give a range of LN users access to an RDBMS

You can use this procedure to give a range of LN users access to a database.

You can restrict the LN user's database authorizations with LN 's Authorization management System (AMS).

Procedure summary

The procedure step and the corresponding session.

- Convert a range of users to a database user - Convert Infor LN User to Database User (ttdba0915m000) . You are prompted to specify the Database Administrator password. Use the Enter Database Administrator Password (ttdba0200s000) session.

RDBMS maintenance procedure

You can use this procedure to maintain the data in the RDBMS Administration module.

The RDBMS Administration maintenance procedure contains these procedures:

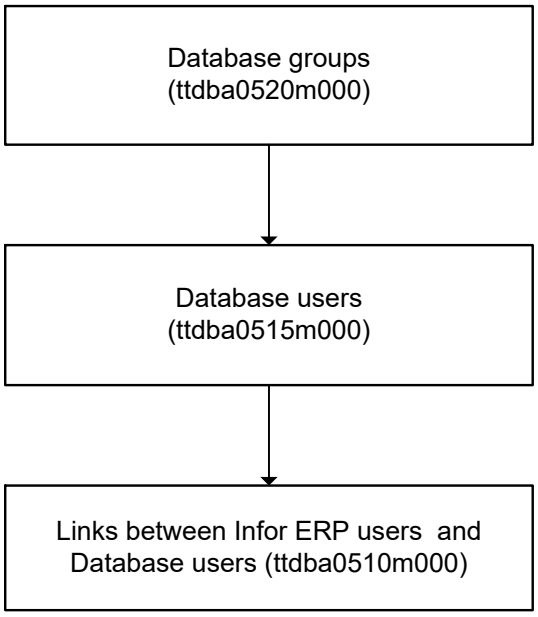
- Create or maintain database groups
- Create or maintain database users
- Create or maintain a link between the LN user and the database user

You can use these procedures as separate and independent procedures. Or as steps in a procedure, which gives individual LN users access to an RDBMS.

This section describes only the most important steps in these procedures.

Note: Some databases cannot handle uppercase characters. Therefore you must create all database user names and group names in lowercase.

This diagram shows the RDBMS Administration maintenance procedure:



To create or maintain database groups

With a database group you can assign a specific set of permissions to multiple users. All members of a database group have the same privileges. A database user must belong to at least one database group. Before you can create a database user, you must create a database group. The reason is that the database user must be assigned to an existing database group. You can use this procedure to create database groups, and to maintain the database group's data.

Procedure summary

The procedure steps and the corresponding sessions.

- Create or maintain database groups

  These sessions are used:

  - Database Groups (ttdba0520m000)
  - Database Groups (ttdba0120m000)
  - Enter Password (ttdba0201s000)
  - Enter Database Administrator Password (ttdba0200s000)

Note: Create all database group names in lowercase.

To create or maintain database users

A database user is a user who has access privileges to specific databases that are stored in the database server. The configuration information for the database user contains all required settings to log on to a database. These settings are automatically loaded when the LN user logs on. The group to which the user belongs determines the access privileges of the database user.

Before you can create a database user, you must ensure that a database group exists. The reason for this is because you must assign an existing database group to a database user. To create or maintain database users, and to maintain the database group's data, you can use this procedure.

Procedure summary

The procedure steps and the corresponding sessions.

- Create or maintain database users

  These sessions are used:

  - Database Users (ttdba0515m000)
  - Database Users (ttdba0115m000)
  - Enter Password (ttdba0201s000)
  - Enter Database Administrator Password (ttdba0200s000)

Note: Create all database user names in lowercase.

To create or maintain a link between an LN user and a database user

An LN user has RDBMS permissions through the link between the LN user and the database user. After you establish the link, the LN user has database permissions for a specific RDBMS.

Before you can use this procedure, you must enure that the LN user and the database user are already defined. If the LN user is not yet created, use the User Data (ttaad2500m000) session to create the user. If the database user is not yet created, use the Database Users (ttdba0515m000) session to create the database user.

Procedure summary

You can use this procedure to give the LN user RDBMS permissions. This procedure links an LN user to a database user. You can also use this procedure to give the LN user permissions for more than one RDBMS.

- Create or maintain a link between an LN user and a database user

  These sessions are used:

  - Links Between Infor LN Users and Database Users (ttdba0510m000)
  - Links between Infor LN Users and Database Users (ttdba0110m000)

Note: You can link a single LN user to a single database user of the same name. Use the Links between Infor LN Users and Database Users (ttdba0110m000) session. For example, you can link many LN users to a single database user. Or link an LN user with a database user of a different name.

RDBMS table and index repository procedure

With this procedure, you can convert the `[RDBMS]_storage` file from the runtime data dictionary to the RDBMS Administration module. During the conversion process, the `[RDBMS]_storage` file is split up into two separate files:

- The database storage parameter file, `[RDBMS]_storage_param`
- The database driver parameter file, `[RDBMS]_driver_param`

If you split up the `[RDBMS]_storage` file, the interaction between LN and the database is improved. For example, the time required to run a query reduces.

Warning: The information in the Table and Index repository affects the data transfer to the database. Ensure to store the correct information in the repository to ensure that you do not corrupt the data transfer.

To optimize the table and index repository:

1 Convert the table and index repository.

a Start the **Convert Table and Index Repository (ttdba0540m000)** session.

b Convert the table and index repository. During the conversion process the `[RDBMS]_storage` file is split up into the `[RDBMS]_storage_param` file and the `[RDBMS]_driver_param` file.

Splitting up the `[RDBMS]_storage` file into two separate files is a once-only action. After you split up the `[RDBMS]_storage` file, the file still exists, but has expired; therefore the file is no longer used.

The next two steps are for maintenance purposes of the `[RDBMS]_storage_param` file and the `[RDBMS]_driver_param` file.

2 Optimize the storage parameters.

a Start the **Storage Parameters Optimization (ttdba0132m000)** session.

b Use this session to maintain the `[RDBMS]_storage_param` file, which is a part of the `[RDBMS]_storage` file.

3 Optimize the driver parameters.

a Start the **Driver Parameters Optimization (ttdba0137m000)** session.

b Use this session to maintain the `[RDBMS]_driver_param` file, which is a part of the `[RDBMS]_storage` file.

RDBMS additional procedures

The RDBMS Administration module contains some additional procedures, which you can use to:

- Change the database group's password and the database user's password.
- Transfer the database group file and the database user file to the RDBMS Administration module.

To change the database group's password

This list shows the procedure steps and the corresponding sessions to change the database group's password:

- Display the database groups using the Database Groups (ttdba0520m000) session.
- Click the **password** buttton for a selected database group using the Database Groups (ttdba0120m000) session.

- Specify de DBA password using the Enter Database Administrator Password (ttdba0200s000) session.
- Change the database group password using the Enter Password (ttdba0201s000) session.

To change the database user's password

This procedure is almost identical to the procedure that you can use to change the database group's password. This section only discusses the differences between the procedures.

The list shows the procedure steps and the corresponding sessions.

- Display the database users using the Database Users (ttdba0515m000) session.
- Click **password** for a selected database user using the Database Users (ttdba0115m000) session.
- Specify de DBA password using the Enter Database Administrator Password (ttdba0200s000) session.
- Change the database user password using the Enter Password (ttdba0201s000) session.

To transfer the database group file and the database user file to the RDBMS Administration module

You usually create the database groups and database users during the LN installation. If the installation was not completed successfully, or partially successful, create the database group file and the database user file in the RDBMS Administration module.

This list shows the procedure steps and the corresponding sessions to transfer the database group file and the database user file to the RDBMS module:

- Transfer Database Group File to RDBMS Administration Module (ttdba0521m000)
- Transfer Database Group File to the RDBMS Admin Module (ttdba0121s000)
- Transfer Database User File to RDBMS Administration Module (ttdba0511m000)

Chapter 10: LN Report Archive

The LN Report Archive solution enables you to archive reports, so you can reprint them, even several years after they are initially printed.

Archiving is particularly useful for external documents, which are reports that are sent, for example, to customers and suppliers, such as invoices and orders.

In some countries, such as in Germany, having this type of reports that can be reprinted for at least 10 years after the reports are initially printed is a legal requirement.

In the LN Report Archive solution, you have these options:

- Specify which reports must be archived. When printed, these reports are automatically stored in an archive folder.
- Reprint the archived reports.
- Maintain the archived reports, for example, move archived reports to another folder or to a CD-R(W)/DVD.

Note: Reports that are printed to Business Objects Reports are archived in the same way as other reports. However, you cannot reprint archived reports on a Business Objects device.

Configuration

Before you can start to archive and reprint reports, you must configure the LN Report Archive. Define the basic parameters for the archive and you must specify which reports must be archived.

To configure the LN Report Archive:

1. Create tables.
2. Define Report Archiving Parameters.
3. Define Report Archive Groups.
4. Specify the reports to be archived.

After you complete the configuration steps, you can check the Report Archiving Rules. You can verify, for various run time settings, whether a report is archived. See "Check Report Archiving Rules".

Create tables

Use the General Table Maintenance (ttaad4100) session to create in company 000 these tables:

- Report Archiving Parameters (ttaad340)
- Report Archive Groups (ttaad341)
- Report Archiving Rules (ttaad342)

Note: The **General Table Maintenance (ttaad4100)** session is password-protected. You can change the password for this session in the Change Password for General Table Maintenance (ttadv0144m000) session. There is no default password.

Create in all companies where reports must be archived the Archived Reports (ttaad345) table. You can create this table also for company 000 to archive the Tools reports.

Note:

To create tables for a company, log on with a user account that belongs to the package combination to which the company is linked.

For example, company 000 is linked to package combination b61a. To create tables for company 000, log on with a user account that belongs to this package combination.

To find out to which package combination a user or a company is linked, use the User Data (ttaad2500m000) and Companies (ttaad1100m000) session is password-protected. You can change the password for this session sessions.

Define Report Archiving Parameters

Use the Report Archiving Parameters (ttaad3140s000) session to define the basic parameters for the report archive. For example, define the directory where the reports must be archived.

Define Report Archive Groups

Use the Report Archive Groups (ttaad3541m000) session to create one or more Report Archive Groups. You can use Report Archive Groups to group reports of the same type, for example, Sales Invoices and Purchase Orders.

In the file system, you can store the archived reports by Report Archive Group: the archive folder specified in the Report Archiving Parameters (ttaad3140s000) session usually contains one or more subdirectories for each Report Archive Group.

This enables you to maintain an overview of the archive and improves the file system performance. Most operating systems begin to respond more slowly if a large number of files are stored in a single directory.

Specify the reports to be archived

Use the Report Archiving Rules (ttaad3542m000) session to define rules that specify which reports must be archived.

In a rule you must specify this information:

- The criteria that determine whether a specific report is archived.
- Link the report to a Report Archive Group.
- How the report must be split when the report is archived.

You can define multiple rules for the same report. For example:

- The Sales Invoice report must always be archived.
- The Sales Invoice report must not be archived if the report is printed in company 755, the test company.

This results in, the Sales Invoice report is always archived, unless the report is printed in company 755.

For details on rules, see the online Help of these sessions:

- Report Archiving Rules (ttaad3542m000)
- Check Report Archiving Rules (ttaad3442m100)

For details on report splitting, see Report splitting on page 114 and the online Help of the Report Archiving Rules (ttaad3542m000) session.

Check Report Archiving Rules

You can use the Check Report Archiving Rules (ttaad3442m100) session to test the effect of the defined Report Archiving Rules. You can, for example, verify for various runtime settings whether a report is archived.

This session is useful if you defined various rules for the same report code.

Example

You defined these rules for the SALES ORDER report:

- All companies/sessions/VRCs: Archiving on.
- Company 100/all sessions/all VRCs: Archiving off.
- All companies/session X/all VRCs: Archiving on.

Use the Check Report Archiving Rules (ttaad3442m100) session to check, whether the SALES ORDER report is archived and printed in company 100, using session X.

In this example, the report is indeed archived because, according to the priority rules, the session is more important than the company number.

For details about the evaluation of report archiving rules, see the online help of the Check Report Archiving Rules (ttaad3442m100) session.

Reprinting archived reports

You can reprint archived reports in these ways:

- Reprint through the Archived Reports session.
- Reprint directly through the Print Archived Reports session.
- Reprint through PDF.

Reprinting through the Archived Reports session

Use this method only if the archived reports are stored in the LN native report format.

To reprint a report:

- Start the Archived Reports (ttaad3545m000) session and search the report that must be reprinted. You can search on properties such as report archive group, keywords, user and print date/time.
- Select the report that must be reprinted and click **Print** to start the Print Archived Reports (ttaad3445m000) session.
- Click **Reprint** in this session to reprint the report.

Reprinting directly through the Print Archived Reports session

Use this method only if the archived reports are stored in the LN native report format, and the properties of the reprinted reports are known.

To reprint reports:

- Start the Print Archived Reports (ttaad3445m000) session.
- Specify the report properties, such as the Report Archive Group and keywords in the selection range fields.
- Click **Reprint** to reprint the reports.

You can print only a limited number of reports in one run.

Reprinting through PDF

Use this method only if the archived reports are stored in PDF format.

To reprint a report:

- Start the Archived Reports (ttaad3545m000) session and search the report that must be reprinted. You can search on properties such as report archive group, keywords, user and print date/time.
- Double-click the report to view the details. Among other things, the full path name of the report file is displayed.
- Open and print the report file through PDF, for example through Acrobat Reader.

Constraints

- You cannot reprint archived reports on a Crystal device.
- You can experience some layout problems if you reprint an archived report on another device then the original report was printed. The problems occur if device properties, such as locale, paper type, and page length, differ for both devices.
- Reprinting through PDF can lead to layout differences because PDF is proportional font based and certain device types are fixed font based.

Maintenance

You can move or delete archived reports and import or export archive settings.

Move/delete archived reports

If the disk on which the report archive is stored is nearly full, you can use the Move/Delete Archived Reports (ttaad3445m100) session.

This session has these options:

- Move a range of archived reports to another folder on the LN server.
- Move a range of archived reports to an offline storage medium, for example, CD_R(W), DVD, or tape.
- Remove a range of archived reports.

See the online help of the Move/Delete Archived Reports (ttaad3445m100) session.

Export/Import archive settings

The settings for the report archive, such as parameters and archive groups, can be exported to and imported from XML files. This is useful, for example, to distribute archive settings to multiple LN servers. You can load predefined archive settings from an XML file that is provided by LN.

- To export archive settings, run the Export Archiving Settings (ttaad3241m000) session.
- To import archive settings, run the Import Archiving Settings (ttaad3242m000) session.

Technical details

This section contains an overview of the components of the LN Report Archive solution. It describes their function in the report archiving and reprinting processes.

Tables

The solution contains these tables:

- Report Archiving Parameters (ttaad340). This table contains general settings, such as the base path for the archive. The table is maintained through the Report Archiving Parameters (ttaad3140s000) session and must be created in company 000 only.
- Report Archive Groups (ttaad341). This table contains the settings, such as extension path and keywords, for the various report groups. The table is maintained through the Report Archive Groups (ttaad3541m000) session and must be created in company 000 only.
- Report Archiving Rules (ttaad342). This table contains the rules that determine which reports must be archived and is maintained through the Report Archiving Rules (ttaad3542m000) session. This table must be created in company 000 only.
- Archived Reports (ttaad345). This table contains details, such as the location of the print file and the report code, on the reports that are already archived. To view these details and reprint archived

reports, use the Archived Reports (ttaad3545m000) and Print Archived Reports (ttaad3445m000) sessions respectively. You must create this table in all companies where reports must be archived.

Sessions

The solution contains these sessions:

- Report Archiving Parameters (ttaad3140s000)
- Report Archive Groups (ttaad3541m000) overview + editable details session
- Print Report Archive Groups (ttaad3441m000)
- Report Archiving Rules (ttaad3542m000) overview + editable details session
- Check Report Archiving Rules (ttaad3442m100)
- Print Report Archiving Rules (ttaad3442m000)
- Archived Reports (ttaad3545m000) overview + editable details session.
- Print Archived Reports (ttaad3445m000)
- Move/Delete Archived Reports (ttaad3445m100)
- Export Archiving Settings (ttaad3241m000)
- Import Archiving Settings (ttaad3242m000)

DLLs

The solution uses these DLLs:

- `ttdllreparch`: A new DLL that is used to check whether a report must be archived and to reprint archived reports.
- `ttstprepdll`: This DLL already existed in Tools and was modified to support report archiving. This DLL splits a print job, depending on the report archiving rule, into various separate files.

See "Report splitting" and the online Help of the Report Archiving Rules (ttaad3542m000) session.

Log file

If errors occur during the archiving of a report, these errors are logged in the `${BSE}/log/log.reprint` log file. Reading the content of this log file is useful if, due to a new report archiving rule, a report is archived for the first time.

Data flow during first print

When running a print command, the Report Archiving Rules (ttaad342) table is checked. The `ttdll reparch` DLL checks whether it contains a rule to archive the report.

If the report must be split, the `ttstprepdll` splits the print job, depending on the split expression specified in the rule.

The split reports are saved as printer-independent files on the location that is specified in these sessions:

- Report Archiving Parameters (ttaad3140s000)
- Report Archive Groups (ttaad3541m000)

That location is: Base path for Archive / Extension path for Archive .

The extension path can contain variables and is evaluated at the moment the files are written.

See "Report splitting" and the online Help of the Report Archiving Rules (ttaad3542m000) session.

Data flow during reprint

If you run a reprint command through the Print Archived Reports (ttaad3445m000) session, you must select a device on which the report must be reprinted.

Then, the `ttdllreparch` DLL searches in the Archived Reports (ttaad345) table for the location of the print files. If the file is found, the report is reprinted on the selected device.

Report splitting

When you print an LN report, a temporary print file is always generated. This file contains the report layout and the data of the records to be printed.

If you print a report that is selected to be archived, the report's temporary print file is split into several smaller print files. They are stored in the report archive folder. The splitting is based on the Split Expression that is defined in the report archiving rule.

Example

The Report Archiving Rules (ttaad342) table contains a rule that indicates that the invoice report must be archived. In this rule, the invoice number is selected as the Split Expression .

You run the Print Invoices session and print a range of 200 invoices. This results in a single temporary print file that contains the layout and the data for 200 invoices. The content of this temporary file is printed.

Because the report must be archived, the temporary file is split into 200 separate files. One for each invoice, which are stored in the report archive folder.

Later on, for example, after nine years, you can reprint any of these 200 invoices individually.

Authorization for archive directories

Archived reports are stored in several subdirectories under the base path that is specified in the Report Archiving Parameters (ttaad3140s000) session. Usually each archive group has its own subdirectories.

See the Extension path for Archive session and the online Help of the Report Archive Groups (ttaad3541m000) session.

You can only archive a report, if you have write authorization for the directory, base path/extension path, in which the report is stored.

To avoid authorization problems, verify that all users have write authorization for all subdirectories under the base path.

Chapter 11: Device management

LN reports can be printed on various types of devices, such as output files and printers. You can set up printers for LN and manage the printer environment.

Raw data is fed into LN and processed into reports, sales invoices, paychecks, and so on. Before you can print the data that LN processes, you must first ensure that your operating system supports the printers in your company.

Before LN carries out a print job, the print job is stored in a temporary file in the `${BSE_TMP}` directory. The printer daemon automatically removes the print jobs after the print jobs are printed. The number of hours between handling a print request and deleting the temporary file is called delete delay time. A record is created for each print request in the **Printer Queue (ttaad320)** table.

The print requests are not automatically removed when the report is printed. Therefore, you must use LN 's device management to purge the printer queue table.

Devices supported by LN

If some physical printers in your company are located in close proximity, you can group the printers in one logical printer. LN 's printer daemon checks the paper types of the printers in the logical printer. It directs the output to the first printer in the logical printer with the paper type that is required for the print job. LN 's device management supports an entire range of device types.

In LN, an output device can be one of these types:

- printer

  A physical printer in a UNIX environment. The print job is sent to the temporary file on the server and queued to the selected record.
- logical printer

  A group of physical printers in a UNIX environment. LN sends the output data to a printer in this group.
- screen

  The output data is shown in a display browser on your monitor.
- append file

  An attachment file in which LN stores the output of a process. If the file does not exist, the file is created automatically.
- rewrite file

  An attachment file in which LN stores recurring data. If the file does not exist, the file is created automatically. The existing files are overwritten.
- direct

A physical printer in a UNIX environment. A device type that you can use if you do not want the printer daemon to handle the printing. At the Device Queue prompt, you must enter the UNIX command to direct the print job to the desired printer.

- windows printer

  A physical printer that Microsoft Windows supports. The client sends the print job to a temporary file on the server. LN 's print tool, BWPrint, then starts and directs the print job to the selected printer. See Windows printer devices on page 121
- windows server printer

  A physical printer that Microsoft Windows supports. The printing is handled by a separate Windows printer server with the Infor ES Reporting Service installed. Users do not require the BWPrint software on their client PCs. See Windows printer devices on page 121.
- **`External Reporting Services`**

  A device to print LN reports through a report design that is stored on a Microsoft Report Server. This Report Server renders the report. The destination of the report depends on the arguments filled in the **Argument for Conversion Program** field in the **Device Data (ttaad3100s000)** session.

  See the *Infor Enterprise Server Plug-in for Microsoft SQL Server Reporting Services Administration Guide*.

In LN 's authorization management, you can define a device preference list in a template. You can link this template to a group of users with the same role in your company.

Convert print output to other formats

You can configure a file device in such a way that the print output is converted into a special format.

To specify the conversion parameters, use these fields:

- Device

  You can choose a unique name for a device. The name SMTP is reserved for Microsoft Outlook to send the print file directly to email. The name FAX is reserved to print directly to a FAX device.
- 4GL program

  Depending on which print output format you want to use, you must specify the name of the conversion script. For example, `ttstpconv`, `ttcmfprint`
- Argument

  You must use this field to specify the print output format. Examples: ascii, html, XML, rtf, XML -PDF.

  To start a program on the client PC to view the print output, certain arguments can be appended with: `[Client program]`. For example; `ascii:excel`, `html:iexplore`, `rtf:winword`.

For details on how to configure a device see the Device Data (ttaad3100s000) session.

Paper types and associated fonts

With LN 's device management, you can define the paper types, fonts, and dimensions of the sheets of paper on which you can print reports. LN can generate various types of output, each of which must be printed on a specific paper type. The paper that is loaded in the physical printers must be suitable for the particular output.

A paper type indicates the characteristics of the paper on which the output can be printed. You can define specific paper types for every type of output. For example, to print paychecks, you must define the characteristics of the preprinted checks that must be loaded in the printer.

Default report settings

The report defaults define a report's default paper type and font. These values are shown in the device selection window, which is displayed when you start the Select Device (ttstpsplopen) session.

Device queue data

Every print request, except the request of printer type Direct, is added to the device queue. LN uses the status of the print job in the device queue to run these actions:

- Restart the print job.
- Preserve the print job.
- Display the print job.

The printer queue data tells you who made the print request, the contents of the request, and the time the request was started up. The data also shows to which the print job is directed and the status of the print request.

Setting up device management

With the LN 's Device Management procedures you can create devices and specify the paper types and fonts for the printouts. Use the additional procedure to group devices in a logical printer.

After the procedure is completed you can use a range of devices to view or print data that LN processes.

Prerequisites:

- Ensure that the physical printers are installed on your operating system. If the printers do not exist on the operating system, see the appropriate Installation Manual for you operating system.
- A printer in an LN environment must have a printer information file or printer device driver. This contains the printer commands, for example, ESC and CTRL. The printer information file also contains the sequences for fonts, reverse video, underlining, and bold.

For detailed instructions on how to specify data in the sessions mentioned in the procedure steps, see the online session Help.

To create the devices and the associated data:

1 Define paper types

Define the characteristics of the paper that can be loaded in a physical printer. You can define specific paper types for every type of output. For example, to print paychecks, you must define the characteristics of the preprinted checks that must be loaded in the printer.

See the Paper Types (ttaad3110m000) session.

2 Define the fonts for the paper type

Define the fonts to use for each paper type. The fonts define the number of characters that can fit on one line. And the number of lines that can fit on a sheet of paper. You must also define the width and height of the sheet of paper. LN uses these dimensions to check whether the report fits on the selected paper type.

See the Fonts by Paper Type (ttaad3111m000) session.

3 Create the devices

You must create the devices to which LN can send processed data. These devices can be physical devices, such as printers or logical printers, or a file.

See the Device Data (ttaad3500m000) and Device Data (ttaad3100s000) sessions.

4 Create the logical printers

You can group printers in close proximity into a logical printer. If you send a print request to a logical printer, the printer daemon checks the paper type of each printer in the logical printer. The output is directed automatically to the first printer in the logical printer that contains the required paper type.

See the Logical Printers (ttaad3101m000) session.

Maintaining device management data

After you have set up the device management, you can print LN 's processed data. LN 's device management gives you the tools to maintain the device management data.

This section describes the most important steps. For detailed instructions on how to specify data in the sessions that are described in the procedure steps, see the online session Help.

To maintain the device queue

If the status of the print job is Waiting, and the device is physical printer, you can maintain the print job data. You can, for example, change the date and time of the actual print, the print job's status, the number of copies, and so on.

After the print job is complete, the print request status changes to *Done*. After the delete delay time expires, the printer daemon automatically removes the temporary file in the `${BSE_TMP}` directory. Although the status of the request changes to *Removed*, the request is not deleted from the printer queue table. A print request with the status *Removed* cannot be displayed or reprinted.

If the status of a request is set to *Preserved*, the output file is not removed if the delete delay time has passed.

Session: Device Queue (ttaad3520m000)

To cancel print jobs

You can only cancel a print job if the status of the print job is *Waiting*, *Active*, or *Being Converted*. You can cancel the print jobs by user or by device. The print request receives the status *Canceled*.

See Cancel Device Requests by User (ttaad3220m000) and Cancel Device Requests by Device (ttaad3230m000) sessions.

To change the device status

You can change the status of a device. For example, when a printer changes location, you can change the status of the printer to *Down*. Even though the printer is down, users can still send print jobs to this printer. The print jobs are queued and are printed when the status returns to *Up*.

Note: You can only change the device status when the Printer Daemon is running.

See the Change Device Status (ttaad3240m000) session.

To purge the device queue data

In the previously described sections, you could change the status of a print request without removing the request from the printer queue. Therefore, the contents of the printer queue table increases in stead of decreases. You must purge the printer queue regularly. Requests with the status *Waiting*, *Active*, or *Being Converted* cannot be deleted. Instead, you must first cancel the requests and then remove them.

See the Purge Device Queue (ttaad3221m000) session.

To change a printer's paper type

You can change the status of a device. For example, when you change the paper in a printer. Check if you must change the printer's paper type.

See the Change Paper Type of Printer (ttaad3102m000) session.

Automatic paper selection for Windows printers

You can configure LN to automatically select the correct paper type when you print a report to a Windows printer device.

Note: This section applies only to devices of type "Windows Printer".

Paper types and paper sizes

In Enterprise Server you can link multiple paper types to each Windows Printer device. You can link each paper type to a Windows paper size code. The Windows printing process uses this code to select the correct setting for the printer.

You can only use paper sizes as defined by Microsoft. See https://docs.microsoft.com/en-us/windows/desktop/Intl/paper-sizes.

Automatic paper type selection

When selecting a Windows printer device in the **Select Device (ttstpsplopen)** session, a check is done. The session checks whether the report fits on the default paper type of the selected printer device. If the report does not fit, the session automatically suggests another paper type that is large enough.

Note:

- This functionality only works if another, large enough, paper type is linked to the selected Windows printer device.
- You can overrule the suggested paper type with another paper type defined for the selected Windows printer device.

Example

The "WINPRINTER" device is linked to the A3 and A4 paper types. The paper types are linked to these Windows paper size codes:

Paper Type	Windows Paper Size Code
A3	A3 (297 x 420 mm)
A4	A4 (210 x 297)

If a report does not fit on the A4 paper type, the **Select Device (ttstpsplopen)** session automatically suggests A3.

Setting up automatic paper selection

This procedure applies only to devices of type "Windows Printer".

- Find out which paper sizes are supported by the Windows printer.
- Link these paper sizes to paper types.
- Link these paper types to the Windows printer.

1 Find out which paper sizes are supported by the Windows printer.

Paper types are supported either by hardware, different trays, or by software.

You can find the supported paper sizes for a printer in the printer's manual, or through the BWPrint preview feature.

To find the supported paper types through the BWPrint preview feature:

a Start a print session, for example **Print Companies (ttaad1400m000)**.

b Specify the selection ranges, and click **Print**. The **Select Device (ttstpsplopen)** session starts.

c Select the Windows printer and select **Preview**. Click **Continue**.

d Right-click in the preview window and on the shortcut menu, select **Paper size > Change device specific paper size**. The **Page Setup** dialog box is displayed.

e You can view the supported paper sizes in the **Size** list.

2 Link paper sizes to paper types

Link each supported paper size you want to use to a paper type.

To link a paper size to a paper type:

a Start the **Paper Types (ttaad3110m000)** session.

b Link the paper size to an existing paper type, or create a new paper type and link the paper size to it. In the **Paper Size** field, you can zoom to the **Windows Paper Size (ttaad3512m000)** session.

Note: the orientation is also important. Many printers have a software option to “Rotate” printing, but there are also printers that have a separate tray for “rotated paper”.

3 Link paper types to printer

Link the desired paper types to the Windows printer:

a Start the **Device Data (ttaad3500m000)** session and double-click the Windows printer. The **Device Data (ttaad3100s000)** details session starts.

b Specify the printer's default paper type in the **Paper Type** field.

c Link the additional paper types to the printer: On the *appropriate* menu, select **Papertype by Device**. The **Papertype by Device (ttaad3107m000)** session starts. Specify the required paper types.

Windows printer devices

You can define devices of type **`Windows Printer`** and **`Windows Server Printer`**.

Both device types use the Windows printing services and drivers. This offers various advantages:

- The Windows printing services have a common set of API's for a wide range of printer devices. A Windows printer driver is available for almost every printer type.
- Infor does not have to deliver printer-specific driver/configuration files with the LN software.
- Barcode printing from LN is widely supported for Windows printers.
- You can use Windows printers to print characters from various character sets on the same report (Unicode printing).
- The Windows API is fully BIDI (bi-directional) enabled.
  - Any printer supported by Windows can print BIDI reports.
  - Windows takes care of mirroring and conversion from logical to display order.
  - Windows fonts can be used to render Hebrew and Arabic characters.

Windows Printer

This device type is used for local printing through both, Web UI and Worktop. To print to a "Windows Printer" device, users require the BWPrint software on their client PCs.

Note: To install BWPrint, the users must install the *LN windows interface* (BW) on their PCs. BWPrint is installed automatically during this installation.

Defining a Windows printer

To create a "Windows Printer" device:

1. Start the **Device Data (ttaad3500m000)** session.
2. Add a device. In the **Device Type** field, select **`Windows Printer`**.
3. Specify the remaining device properties. See the online help of the **Device Data (ttaad3100s000)** session.

Windows Server Printer

This device type provides the possibility to print reports through a Windows driver. This enables you to reroute printing of reports from UNIX-based LN environments to a Windows system. This is particularly of interest for reports that are related to batch processes. It reduces the requirement for LN-related UNIX device drivers.

Windows Server Printers can print in multiple ways:

- Through the Infor ES Reporting Service.
- Through the Cloud Printing Service.
- To a local printer.

Defining Windows Server Printers that print through the Infor ES Reporting Service

In an on-premises installation on a UNIX platform, Windows Server Printers can print through the Infor ES Reporting Service. The Infor ES Reporting Service must be installed on a separate Windows printer server.

To define a Windows Server Printer that prints through the Infor ES Reporting Service:

1. Configure the printer server.

   Install the Infor ES Reporting Service on the printer server. During the installation, you must assign a port number to the Infor ES Reporting Service. See the *Infor LN Installation Guide*.
2. Define the printer server as a remote system.
   a. Start the **Systems (ttaad0550m000)** session.
   b. Add a record that represents the printer server. In the **Service Type** field, select **`Infor ES Reporting Server`**. See the online help of the **Systems (ttaad0150s000)** session.
3. Define a Windows Server Printer.
   a. Start the **Device Data (ttaad3500m000)** session.
   b. Add a Windows Server Printer. Points of attention:
      - In the **Device Type** field, select **`Windows Server Printer`**.
      - In the **Argument for Conversion Program** field, specify this text:

        `server=[the name of the printer server, as defined in the` **`Systems (ttaad0150s000)`** `session] port=[the port number that is assigned to the Infor ES Reporting Service]`

        For example: `server=cnl10050 port=7688`

See the online help of the **Device Data (ttaad3100s000)** session.

Defining Windows Server Printers that print through the Cloud Printing Service

Windows Server Printers can print through the Cloud Printing Service in these installation types:

- On-premises installation on a UNIX or Windows platform
- Cloud installation

The Cloud Printing Service can print to a printer or to a fileshare.

On-premises installation on a UNIX or Windows platform

To define a Windows Server Printer that prints through the Cloud Printing Service:

1. Install the Cloud Printing Service on the printer server. See the *Infor Cloud Printing Service Administration Guide*.
2. Start the **Device Data (ttaad3500m000)** session.
3. Add a Windows Server Printer. Points of attention:
   - In the **Device Type** field, select **`Windows Server Printer`**.
   - In the **Argument for Conversion Program** field, specify this text:

     `-cloudPrinting`
   - In the **Device Queue** field, specify the appropriate printer queue or fileshare:
     - For a printer queue, specify the UNC path in this format: `\\servername\printername`
     - For a fileshare, specify the UNC path in this format: `\\servername\...\folder`.
4. Add the printer or fileshare in the Cloud Printing Service. See the *Infor Cloud Printing Service Administration Guide*.

Cloud installation

To define a Windows Server Printer that prints through the Cloud Printing Service:

1. Install the Cloud Printing Service on the printer server. See the *Infor Cloud Printing Service Administration Guide*.
2. Start the **Device Data (ttaad3500m000)** session.
3. Add a Windows Server Printer. Points of attention:
   - In the **Device Type** field, select **`Windows Server Printer`**.
   - In the **Device Queue** field, specify the appropriate printer queue or fileshare:
     - For a printer queue, specify the UNC path in this format: `\\servername\printername`
     - For a fileshare, specify the UNC path in this format: `\\servername\...\folder`.
4. Add the printer or fileshare in the Cloud Printing Service. See the *Infor Cloud Printing Service Administration Guide*.

See the online help of the **Device Data (ttaad3100s000)** session.

Defining Windows Server Printers that print to a local printer

In an on-premises installation on a Windows platform, Windows Server Printers can print to a local printer. For example a printer that is connected to a TCP/IP port.

To define a Windows Server Printer that prints to a local printer:

1 Start the **Device Data (ttaad3500m000)** session.
2 Add a Windows Server Printer. Points of attention:
   - In the **Device Type** field, select **`Windows Server Printer`**.
   - In the **Device Queue** field, specify a printer that is accessible for all users.

     For example, specify a local OS printer that is connected using a Standard TCP/IP port and that is available on the network.

     Note: Do not specify a "network printer" because a "network printer" is not accessible for all users. A "network printer" is only visible in the Windows user profile of the user that defined the "network printer".

See the online help of the **Device Data (ttaad3100s000)** session.

Microsoft Excel devices

You can define devices to print reports to an MS Excel spreadsheet.

Note: Printing reports to an MS Excel device is not supported in Worktop.

Defining a Microsoft Excel device

To create a Microsoft Excel device:

1 Start the **Device Data (ttaad3500m000)** session.
2 Add a device. The **Device Data (ttaad3100s000)** session starts.
3 Complete these steps:
   a Specify the name and description for the device.
   b In the **Device Type** field, select **`External Reporting Services`**.
   c Leave the **Conversion Program** empty.
   d In the **Argument for Conversion Program** field, specify `-excel`, optionally followed by other parameters. For example:
      - `-excel`

        Print the data to an Excel file. The file is stored in a folder on your computer. This folder is specified in the **Path** field. The file name has this format: `[report name]_[company number]_[yyyymmdd-hhmmss].xlsx`

        This is a sample file name: `ttaad340001000_000_20130626-202842.xlsx`
      - `-excel -open`

        Similar to `-excel`. The Excel file is automatically opened.
      - `-excel -nodatetimestamp`

        Similar to `-excel`. The Excel file has no date- and timestamp.

        The file name has this format: `[report name]_[company number].xlsx`

        This is a sample file name: `ttaad340001000_000.xlsx`
      - `-excel -server`

Similar to `-excel`. The file is stored in a folder on the server. This folder is specified in the **Path** field.

e In the **Path** field, specify the folder where the Excel file is stored.

Note:

- If you use the `-server` option in the **Argument for Conversion Program** field, specify a folder on the LN server. For example, `/home/jdoe/tmp` (Unix).
- If you do not use the `-server` option in the **Argument for Conversion Program** field, specify a folder on your computer. For example, `c:\temp`.
- If you use LN UI, the Excel file cannot be copied to your client computer. In that case, the file is transferred through a download action.

f Specify the remaining device properties and save the device. For details, see the online help of the **Device Data (ttaad3100s000)** session.

Printing reports to a Microsoft Excel device

To print a report to a Microsoft Excel device:

1. Start the print session to which the report is linked.
2. Print the report.

   Points of attention:

   - In the **Device** tab in the print session, select **`Advanced Device`**.
   - In the **Printer** tab in the **Select Device (ttstpsplopen)** session, select the Microsoft Excel device.

   A flat file with all input fields of the report is generated.

Session-based reporting

Most reports in LN are 4GL reports. These reports consist of a layout part and, optionally, a report script. You can edit these report layouts in the classic report editor.

See the *Infor LN Development Tools Development Guide*.

LN supports Report Viewer enabled reports. When printed, these reports are displayed in the LN Report Viewer. Developers can edit these reports in the Reporting Studio Report Designer. The Report Designer offers various useful features to beautify the reports.

You can convert existing 4GL reports to Report Viewer enabled reports. These converted reports are called session-based reports. How to set up session-based reporting and how to print session-based reports is described later.

1. Convert the reports to XML report designs
   a Start the **Enable Reports for RPT Design (ttadv3241m000)** session.
   b Specify the required information and click **Convert to Runtime**.

   For details, see the session help.

Note:

- `.rptdesign` file name extension. The generated report designs are stored in the report sources folder and have a
- Developers can transfer the report designs to their client PCs and beautify the layout of the converted reports in the Reporting Studio. For example, developers can add a graph or a company logo to the report layout.

  See **To create session-based reports** in the *Infor LN Development Tools Development Guide*.
- To change the structure of a session-based report, for example to add layouts, input fields, or labels, developers must use the classic report editor.

  See the *Infor LN Development Tools Development Guide*.

2 Create a Report Viewer device

To print session-based reports, a device is required that sends the report output to the LN Report Viewer.

To create a Report Viewer device:

a Start the **Device Data (ttaad3500m000)** session.

b Add a device with these properties:

- **Device**: RPT
- **Description**: Report Viewer
- **Device Type**: Rewrite File
- **Locale**: <empty>
- **Intermediate File in XML Format**: Yes
- **Driver**: <empty>
- **Shell Command**: <empty>
- **Conversion Program**: <empty>
- **Argument for Conversion Program**: XML -RPT
- **Path**: ${HOME}/rpt

3 To run session-based reports

To print a session-based report to the LN Report Viewer:

a Start Web UI and log onto your LN server.

b Start the print session to which the report is linked.

c Specify the required selection ranges and print options, and click **Print**. The **Select Device (ttstpsplopen)** session starts.

d Select a Report Viewer device and click **Continue**.

Note:

- All business logic that is defined in the session and the original 4GL report is executed. Therefore, the report can be part of the business process. For example, the report can perform transactions in the database or change the status of a record.
- The LN Report Viewer only works with Web UI. Worktop users cannot print Report Viewer enabled reports.
- From the Report Viewer, you can send a report to, among other things, a printer or a PDF file.

Using the Report Viewer

The Report Viewer is used to display session-based and query-based reports.

In the Report Viewer you can perform these actions:

- Navigate through the report data.
- Export report data.
- Print the report.

Toolbar buttons

This table shows the available toolbar buttons:

Button	Name	Description
	Export data	Starts the **Export Data** dialog box, where you can select report columns and meta data, such as layout names and data types, to be exported. The data is exported in csv (Comma Separated Values) format. You can store the data in a file or view it directly, for example in MS Excel.
	Export report	Starts the **Export Report** dialog box, where you can select these options: • The output format: Excel, PostScript, PDF, Word, or PowerPoint. • The pages to export. • The report size. You can store the data in a file or view it directly.

Button	Name	Description
	Print report	Starts the **Print Report** dialog box, where you can select these options: • The output format: HTML or PDF. • The report size, only for PDF. • The pages to export. The output is displayed in a print preview window. If you selected HTML, you are automatically prompted to select a printer. If you selected PDF, click **Print** to print the output.
	First page	Navigates to the first page.
	Previous page	Navigates to the previous page.
	Next page	Navigates to the next page.
	Last page	Navigates to the last page.
	Go to	Navigates to the page that is specified in the **Go to page** field.

Microsoft Reporting for LN

You can create a device of type **`External Reporting Services`**.

This device can print LN reports through report designs that are stored on a Microsoft Report Server. This report server renders the reports.

The destination of the reports depends on the arguments filled in the **Argument for Conversion Program** field in the **Device Data (ttaad3100s000)** session.

See the *Infor Enterprise Server Plug-in for Microsoft SQL Server Reporting Services Administration Guide*.

Document Output Management

This feature provides electronic distribution of documents and reports to recipients. Recipients can be any known party in the LN environment such as business partners, employees, or system users.

You can specify preferences by document type, by distribution channel, and by recipient. For example, to distribute invoices you can distribute one copy through email to one business partner and a paper copy to another business partner.

For details about the setup and usage, see the *Infor LN Document Output Management User Guide*.

Chapter 12: Text management

You can use text in LN 's Text Management for various purposes.

LN 's text management provides the tools to write and maintain text in LN. You can use LN 's text editor, for example, to write queries in LN 's SQL Queries module. You can also provide information about the data stored in the database tables.

You have these options:

- Link information to a record that defines the contents of the record.
- Give additional information, for example, about an item or sales order.

You can also use LN 's text management to create the text-related templates that define the user's text authorizations. You can define the text-related authorizations in templates, which you can link to the users with LN 's User Management module.

Note: The text in LN is typically written in the multiline editor. Although the Infor ERP editor is still supported, the multiline editor, which is more user-friendly, is preferred.

Description of text-related templates

A normal user must have some basic text authorizations. You can define these text authorizations for the user:

- Use

  The user is authorized to read texts and link the text to a record, for example, an order, item, and so on. The user cannot change the texts.
- Update

  The user is authorized to create, change, and read text. The user is also authorized to link the text to a record.
- Read

  The normal user can only read texts for the specified text group. The user has no authorization to link or update texts.

The required data and authorizations are defined in the text parameters. These text parameters are defined in these templates:

- Text group authorization template
- Default text groups template
- Default text groups by text field template

For each text group, you must specify the text authorizations, which define the read, use, or update authorizations in a template. If a text is written in a text field for which no text group is defined. The default text group is used that is defined for the specified text field in the text groups by text field template. If no default text group is defined for the specified text field, LN uses the information that is defined in the default text group template.

You can specify text parameters either for a specific company or for all companies. The text parameters that are defined for a specific company take precedence over the defaults that are defined for all companies.

Text management parameters

Use the text parameters procedure to define the basic parameters that are required to use text in LN. These parameters provide the users with the basic requirements to write and edit text.

After completing the procedure the users have the authorization and means to create and edit text in LN

No prerequisites are required for the environment.

1 Create text windows

You must specify the text window's properties that define the layout of the text window. For example, the text window's frame, color, size and shape.

If you use the multiline editor as the external editor, only the number of columns and number of rows are relevant. In that case, the frame and color are defined by default. You must make a selection in all fields because you can also use other editors for the text group.

See the Text Windows (tttxt1520m000) and Text Windows (tttxt1120s000) sessions.

2 Create text groups

You must define the text group's properties. For every type of text, you can define text groups with specific properties. You must assign the text editor best suited to the text window's function. For example, you can use a multiline editor to write queries. Or use Microsoft Word for texts that contain additional information, for example, about items and sales orders.

See the Text Groups (tttxt1110m000) session.

3 Create the template for the text group authorizations

In the text group authorizations template, you must specify the user's authorization to edit the text in LN. For example, you can grant the users who are linked to the template permission to read, use, or update text. The text group authorization template is linked to the user in the User Data (ttaad2500m000) session.

See the Text Group Authorization Template (ttams1122m000) session.

4 Create the default text groups by text field template

In the default text groups by text field template, you must specify the text group for a specific text field. If you do not specify a default text group for a text field, LN uses the template that is defined in the default text group template:

See the Default Text Groups by Text Field Template (ttams1120m000) session.

5 Create the template for the default text groups

In the default text groups template, you must specify the text group that LN must use if no text group for a text field is defined.

See the Default Text Groups Template (ttams1121m000) session.

Text management maintenance

The most important steps in the text management procedures are described here.

Instructions on how to specify data in the sessions that are described in the procedure steps, see the online session Help.

To create a range of text numbers

You can use the text numbers to exchange text with LN's Exchange module between various locations or sites. Ensure that, for each site, you define unique ranges for the text groups to avoid conflicts with identical text numbers on separate sites.

See the Text Number Range by Text Group (tttxt0130m000) session.

To remove unused texts

Unused texts are texts that are no longer linked to a record.

A text becomes unused in these situations:

- You remove the record, for example, the sales order or the item, to which the text was linked.
- You unlink the text from a record, using the `Disconnect` command in the Text Editor.

Removing the unused texts occasionally is useful to avoid pollution in the database.

See the Delete or Print Unused Texts (tttxt1210m000) session

Chapter 13: Job management

You can use job management in LN to schedule jobs that are based on your organizational requirements.

For example, you can schedule jobs at non-peak hours to improve the overall system performance in a heavily loaded environment. A job consists of one or more sessions or shell commands, or both, that run without user interaction. The sessions and shell commands in a job can be started while you are not logged on to LN. You can schedule jobs to start processes periodically, at a defined interval, or immediately. Typically, you use LN job management for print and processing sessions.

Job data

To create a job, you must specify basic job data and link sessions or shell commands, or both, to the job.

In the basic job data you specify whether the job is periodical. For periodical jobs, you specify how the job is scheduled. You can use a job to run one or more sessions:

- Once, at a predefined time
- Repeatedly, at specific times
- Periodically

You can use the job management calendar to define the date and time the job must be started. Using the job calendar, you can start the job during quiet periods. For example, you can schedule processing and printing jobs to run after office hours to improve the system performance during the day.

In LN, you can create these two job types:

- Periodical jobs that are carried out multiple times

  For a periodical job, you can specify a time interval or a calendar is specified. You can also use the advanced scheduling options to determine when the job must run. If desired, you can restart the job automatically.
- Non-periodical jobs that are carried out only once

  When the job is carried out, the job data is removed automatically.

See Job data procedure.

Job alerts

In the **Job Data (ttaad5100s000)** session, you can define a job alert for these job events:

- Job has started
- Job has been executed successfully
- Job has failed with runtime error
- Job has been executed with runtime errors
- Job has been skipped
- Job has been postponed
- Job has been canceled manually
- Job has been canceled due to timeout
- Job is running longer than … % of the expected duration
- Job is running longer than hh:mm

Job alerts are published to ION through the `LnMessage` BOD. In ION, the LN_Message_Job_Alert monitor is available, which can be used to distribute job alerts. A drillback to the **Job Data (ttaad5100s000)** session is supported. See the *Infor LN Integration Guide for Infor ION Workflows and Monitors*.

Before you can use the job alerts feature, you must enable it. To do so, select the **Enable Job Alerts** check box in the **Tools Parameters (ttaad0100m000)** session.

Note: This check box is only available if the `LnMessage` BOD is supported in company 0. If the check box is not available, an update of the LN application is required.

Advanced scheduling options

If the scheduling options using a calendar or running the job at fixed intervals is not sufficient, the job scheduling lines can be used.

The job scheduling lines can have these forms:

- Every

  Run the job in a fixed time interval, every number of minutes/hours within the day. Days can be excluded.

  Example: A picklist for the warehouse must be created every 15 minutes on workdays during working hours. Two Every schedule lines can be created. One to run from 08:00 till 12:00. One from 13:30 till 18:00 for every 15 minutes including the start time (08:00 and 13:30), excluding end time (12:00 and 18:00). Except Saturday and Sunday.

  Note: An Every schedule starts at the start time of the schedule. If the picklist runs on 11:45, then the next time in this schedule is 13:30, the start time of the second schedule line.
- Daily

  Run the job every fixed number of days on a fixed time. A ‘Valid from’ date is required and is used as a reference to calculate which day is the next day in the schedule.
- Weekly

Run the job every fixed number of weeks on a fixed time on selected days of the week. A ‘Valid from’ date is required. It is used as a reference to calculate which week is the next week in the schedule. The day of the valid from date is also seen as the start day of the week. If the valid from date is Wednesday July 8th 2020 and Monday is a selected day. The first Monday in the schedule is Monday July 13th.

- Monthly

  Run the job on a fixed time on fixed dates or relative days in selected months. The schedule can be the 10th, 20th and 30th of January, May, August and December (dates). Or the second and last Saturday of all months (days).

If no next execution date can be calculated, either because the calendar is expired, then the status of the job is set to *Blocked* with a next execution date of 9999-12-31.

To validate the defined schedule the Calculate execution dates for a job (ttaad5408m000) session can be used. This session calculates all execution dates for a selected job, starting with the start date and time until the next date after the entered end date.

Shared job data tables

Typically, each company stores its own basic job data. Therefore, a job runs for a particular company.

In a job, you can also run sessions in more than one company. You can run sessions in multiple companies when the job data tables of the associated companies are physically mapped to a single main company. You can use the Logical Tables (ttaad4120m000) session to link these job tables as logical tables to one physical table:

- **Job Data (ttaad500)**
- **Sessions by Job (ttaad501)**
- **Job Input Variables (ttaad503)**
- **Job Daemon Indicator (ttaad505)**
- **Calendars (ttaad506)**
- **Dates by Calendar (ttaad507)**
- **Job Scheduling Lines (ttaad508)**
- **Job History (ttaad510)**
- **Job Session History (ttaad511)**
- **Job Session History (ttaad512)**

Before you add a session to a job, you must check the company number. If a job applies to more than one company, you must switch to the other companies with the **Change Company** command. Define the job for each company. You can then define the job data for the other companies.

Job execution

Jobs can be started in multiple ways. The job's status defines how you can start the job. You can start the job if the job's status is **In Queue** or **Free**.

In Queue status

If the job's status is In Queue, you can use the Job Scheduler BSE Service to start the job. Ensure the Job Scheduler BSE Service is running before it can activate any job.

Jobs with status In Queue can be monitored using the Scheduled Jobs (ttaad5120m000) session. This session shows an overview of all scheduled jobs across all companies and displays the this information:

- Company: The company in which the job is created.
- Job code
- User: The owner of the job.
- Next Execution: Shows the date/time when the job will be started.
- Status: The status can be:
  - Waiting
  - Starting
  - Running
  - Failed to Start

    For jobs with the *Failed to Start* status, an error is logged. After that, a new attempt is made to start the job.
- Hostname: The name of the server where the job is running.
- Bshell ID: The process ID of the bshell starting the job, when the job is running.

Jobs started by the Job Scheduler BSE Service are carried out by a separate bshell. This bshell runs in the company of the job, using the settings of the LN user who created the job.

Job Daemon

In case BSE Services are not used, you can use the Job Daemon to start the job. Ensure the Job Daemon is running before it can activate any job.

The implementation of the Job Daemon depends on the operating system of the LN server:

- On a UNIX server

  The Job Daemon is the shell program `rc.startjobdm`, which is located in the `${BSE]/etc` directory. To start the Job Daemon, you can include the `${BSE}/etc/rc.startjobdm` command in the `${BSE}/etc/rc.start` file. Therefore, the Job Daemon is activated when you start the LN environment.
- On a Windows server

  The Job Daemon is a service that you can manage with the Application Service Manager (ASM)

When the Job Daemon runs, you can stop your bshell without any problem. The Job Daemon, and not your bshell, carries out the job.

To run a job for multiple companies, you must start a Job Daemon for each company.

See Job handling procedure.

Free status

If the job's status is Free, you can start the job in these ways:

- Use the Activate Job (ttaad5203m000) session
- Use the `rc.startjob` shell command (UNIX platform)
- Use the startjob.bat batch file (Windows platform)

Activate Job (ttaad5203m000)

If you start the job before the time and date of execution, the job's status changes to **Waiting**. When the job is being carried out, the job's status changes to **Running**. Because the bshell of the user who starts the job carries out the job. The user cannot quit the bshell until the job is finished.

To start a job that belongs to more than one company, you must use the **Change Company** command to switch to the required company.

The rc.startjob shell command (UNIX)

You can also start the job with the UNIX `${BSE}/etc/rc.startjob` shell script. Specify this syntax at the UNIX prompt:

```
${BSE}/etc/rc.startjob [job name]
```

You can include the `rc.startjob` script in a UNIX user job scheduler (crontab), which can automatically start jobs. For this purpose, some UNIX variables must be defined in the shell script. For more information, see the shell script.

To run a job for multiple companies, you must create a unique version of `rc.startjob` for each company.

The startjob.bat batch file (Windows)

If LN runs on a Windows server, you can also start the job with the startjob.bat batch file.

To start a job, specify this syntax at the Windows command prompt: `startjob.bat [job name]`

Before you can specify this command, you must run the `fillenv.bat` batch file to set the LN environment variables.

To automate the execution of jobs, you can run `startjob.bat` with a scheduler.

To run a job for multiple companies, you must create a unique version of `startjob.bat` for each company.

See Job handling procedure.

Job status

This table shows the statuses a job can have during its life span:

Status	Description
Free	The job's basic job data is defined and is ready to be started.

Status	Description
Running	The job is started and is running.
Waiting	The job is started and is waiting to be carried out at the next execution date and time. The Job Daemon does not start the job.
In queue	The job is queued and is started automatically when the Job Daemon runs.
Blocked	The job is blocked and cannot be started. To start the job, set the status to Free or In Queue.
Canceled	The job is stopped with the Cancel Jobs (ttaad5204m000) session. A job is also canceled when the maximum duration is exceeded.
Runtime error	This status signals an error during the execution of the job.

If required, you can cancel a job that the Job Daemon starts, for example, to change the job's basic job data. To cancel the job, use the Job Data (ttaad5500m000) session. On the *appropriate* menu, click **Cancel Job**. If you cancel a job, the current session of the running job is carried out, but the next sessions in the job are not started. The job's status changes from **In Queue** to **Canceled**. To restart a canceled job, you must release the job to change the job's status to **In Queue** or **Free**.

To temporarily stop a job, start the Job Data (ttaad5500m000) session. On the *appropriate* menu, click **Block Job**.

Job groups

To handle dependent jobs, you can create a job group. Job groups are convenient when running job 'A' depends on the result of job 'B' and you do not know at what time job 'A' must be scheduled.

This session is a multi-main table session. It shows multiple tables and you can perform maintenance commands on the main entity and the child tables.

You can change the status of the job group with the specific options in the **Job Groups (ttaad5140m000)** session and the **Job Group Data (ttaad5640m000)** session. The statuses have the same meaning as the statuses of the Jobs and the same status changes are allowed. Job groups are only handled by the BSE service. You cannot activate Job Groups directly.

Tip: Jobs that are started through a Job Group must be in status "Free".

Creating a job group:

1. Start the **Job Groups (ttaad5140m000)** session to create a Job Group.
2. Click **New**.
3. Specify a Job Group name with the same characteristics as a Job name.
4. Specify a Job Group description. The **Job Group Status** field and the Job Group **User** field are specified automatically.

5 Click **Add** in the **Jobs by Job Group (ttaad5143m000)** tab to add jobs to the group.

Add the jobs in the order they must be performed. The first Job in the group, the one with the lowest Job Group Number, determines the handling of the whole Job Group. For example, the executing date and if the Job Group must be repeated. If the Job Group is not repeating the Job Group and all non-repeating Jobs in the Job Group are deleted when the Job Group is finished. In the **Action on Error** field you can specify what happens when a job ends in an error.

You can use these states:

- Continue

  Makes the job group start the next session(s).

- Interrupt

  You can appoint a last job in the group that must be performed, to clean up. The job group stops processing after performing that job. All other jobs that are not scheduled are not started.

6 Click the **Jobs by Job Group Dependencies (ttaad5146m000)** tab to determine the dependencies.

A job in the Job Group can only depend on a job in the same Job Group with a lower job number. You can only use Job Groups with the BSE Service **Job Scheduler Service (ttaad5220m000)** session.

Job history

When the execution of a job stops, information is written to a history log. This happens, for example, when the job completes successfully or when a runtime error occurs.

The job history contains information about the date and time of the execution and the reasons why the job and its associated session has ended. Use the Print Job History (ttaad5411m000) session to print this history.

During the execution of a job, messages are suppressed. These suppressed messages are also logged in a history. You can use the Job History Messages (ttaad5512m000) session to print these messages.

If you start jobs regularly, you must purge the job history list and the message history list regularly with the Delete Job History (ttaad5210m000) session.

Defining downtime

In the **Planned Downtime (ttaad5530m000)** session, you can define a planned downtime period.

During planned downtime neither users nor jobs should use the system. You can plan your work in such a way that no long running tasks or jobs are started before the downtime. Running jobs are killed when the system is brought down and this can cause issues.

To define the downtime period, you must specify this information:

- **Upgrade Date/Time**

Specify the start date/time of the downtime period. Jobs are not started if they, based on their expected duration, would still be running at the start of the downtime period.

The calculation of the expected duration of a job is based on the recorded history of that job. Only the history with status (‘cause’) ‘Executed’ is taken into account because these records give the total time required to successfully run the job.

- **End Period**

  Optionally, specify the end date/time of the downtime period. If you specify the **End Period**, jobs are also not started between the **Upgrade Date/Time** and the specified end time.

After the downtime period, the jobs that should have started during the downtime period are automatically scheduled to start directly.

Job data procedure

After completing the procedure a job is ready to be started.

No prerequisites are required for the environment.

Before you can use periodical jobs in LN, you must define the basic job data. In LN, you can start periodical jobs according to a schedule that is defined in the job calendar, a regular interval or using job scheduling lines. You can define the interval in the basic job data. After you define the basic job data, you can add sessions and shell commands to the job. The default settings of the sessions in the job can be maintained, when required, with an optional step in the procedure.

1 Create a job calendar.

To start periodical jobs, you can use a job calendar. The job calendar contains the dates and times the job must be carried out.

See the Calendars (ttaad5106m000) session.

2 Specify the dates in a job calendar.

In the job calendar, you must define the date and time that the job must be started.

See the Dates by Calendar (ttaad5107m000) session.

3 Define the basic job data.

First define the basic parameters for the job, before you can add sessions to a job with the Add Session to Job (ttaad5102s000) session. You can make a periodical job non-interactive. To start the job, specify a predefined job calendar, a regular interval or scheduling lines in these sessions:

- Job Data (ttaad5500m000)
- Job Data (ttaad5100s000)

If you select the **Suspend Job until Next Execution** check box, the job is suspended until the next execution date. When jobs are started by the Activate Job (ttaad5203m000) session, the UNIX shell program `rc.startjob`, or the Windows batch file `startjob.bat`. The job’s status is set to **Waiting**. The job is automatically restarted at every subsequent execution date that is defined in the calendar.

If you clear the **Suspend Job Until Next Execution** check box, the job is carried out once and then terminated. When jobs are started by the Activate Job (ttaad5203m000) session, the UNIX shell program `rc.startjob`, or the Windows batch file `startjob.bat`. The job's status is set to **Free**. To restart the job, you must manually start the Activate Job (ttaad5203m000) session, the UNIX shell program `rc.startjob`, or the Windows batch file `startjob.bat`.

4 Add sessions to a job.

A job must contain one or more sessions. These sessions must be carried out according to the data that is specified in the Job Data (ttaad5100s000) session. Typically, a job starts print or process sessions.

Under Error Handling, you can specify how LN must continue if an error occurs during the execution of the session. You can, for example, indicate that LN must execute, or skip, several sessions of the job if an error occurs during the execution of the session.

5 Optionally, add a shell command to the job.

You can use this session to link the job to a shell command to a job to manipulate the command's execution. To ensure that the shell command has the correct effect on the job, you must test the script. Typically, to test the job and shell script, you can redirect the output to a file or printer. You can redirect the output to a file or printer with one of the operating system's output redirection facilities.

See the Add Shell Command to Job (ttaad5205m000) session.

6 Change the default settings of the sessions in the job.

This procedure is optional, and you can use this procedure to change the settings of the sessions that are linked to the job. For example, you can change the execution order of the sessions or the device to which the output of a session is sent.

See the Sessions by Job (ttaad5501m000) session.

Job handling procedure

With the job handling procedures, you can start, block, cancel, queue, and release jobs.

This procedure results in a job that runs without user interaction according to one of these specifications:

- The job calendar.
- The interval that is defined in the job's basic data.
- The scheduling lines.

This helps you to use LN 's resources more efficiently.

Note: You can only use the job handling procedure when the job data procedure is completed.

Steps in the job management procedures

Depending on the job's status, these methods are available to start a job:

- If the job status is *In Queue*, the Job Scheduler BSE Service, or the Job Daemon, starts the job.

If you use the Job Scheduler BSE Service, the job starts automatically and a running bshell is not required. The Job Scheduler BSE Service starts a bshell in the background, which starts the job with the **Activate Job (ttaad5203m000)** session.

You must start the Job Scheduler BSE Service to ensure that the Job Scheduler BSE Service can start any job. If the Job Scheduler BSE Service runs, you can exit the bshell without any problem. The Job Scheduler BSE Service, and not your bshell, carries out the job.

- If the job's status is *Free*, you can start the job manually with these options:
  - The **Activate Job (ttaad5203m000)** session
  - The UNIX shell program `rc.startjob`
  - The Windows batch file `startjob.bat`

    In this case, you must have a running bshell.

To enable and start the Job Daemon on a Windows platform

Note: We recommend that you use the Job Scheduler BSE Service instead of the Job Daemon. For more information see BSE Services on page 144.

The Application Services Manager (ASM) manages (externally) the Job Daemon service.

Typically, a Job Daemon runs jobs for one company. To run jobs for more companies, you must create a Job Daemon entry, in ASM, for each company. Per entry, use the `BSE_COMPNR` variable to specify the company for which a Job Daemon must be started. Use the `PACKAGE_COMB` variable to specify the package combination to which that company is linked.

To start the Job Daemon on a UNIX platform

Note: We recommend that you use the Job Scheduler BSE Service instead of the Job Daemon.

To start the Job Daemon, you can include the `${BSE}/etc/rc.startjobdm` command in the `${BSE}/etc/rc.start` file. Therefore, the Job Daemon starts when you start the LN environment.

Typically, a Job Daemon runs jobs for one company. To run jobs for multiple companies, you must change the `rc.start` file to start Job Daemons for each company. Per Job Daemon entry, use the `BSE_COMPNR` variables to specify the company for which a Job Daemon must be started. Use the `PACKAGE_COMB` variable to specify the package combination to which that company is linked.

Stop the Job Daemon

To stop the Job Daemon you can include the `${BSE}/etc/rc.stopjobdm` command in the `${BSE}/etc/rc.stop` script. Therefore, the Job Daemon stops when you stop the LN environment. To stop jobs for multiple companies, you must include multiple entries of `rc.stopjobdm` in the `rc.stop` file. Use the `BSE_COMPNR` variable to specify the company for which a Job Daemon must be stopped. Use the `PACKAGE_COMB` variable to specify the package combination to which that company is linked.

To run a job with the Job Daemon

To use the Job Daemon to start a job, you must ensure that the job's status is In Queue.

See the Job Data (ttaad5500m000) session.

To run the job with Activate Job (ttaad5203m000)

Because your running bshell starts the job, you cannot close the bshell until the job is finished. If the job is started before the time and date of execution, the job's status changes to Waiting. When the job is carried out, the job's status changes to Running.

You can start a job that belongs to more than one company. Switch to the required company with the **Change Company** command, before you start the job.

To run the job with the rc.startjob shell program

You can also start the job with the UNIX shell program `${BSE}/etc/rc.startjob`.

To start the job, at the UNIX prompt run this command::

```
${BSE}/etc/rc.startjob [job name]
```

Scheduling

If the `rc.startjob` program is included in a UNIX crontab, the program can automatically start jobs. For this purpose, some UNIX variables must be defined in the shell script. Additional information about how to define the UNIX variables is included in the shell script of the `rc.startjob`.

Run jobs for multiple companies

If you must start jobs that pertain to different company numbers, different versions of the `rc.startjob` script are created.

To run the job with the startjob.bat batch file

On a Windows platform you can start jobs with the `startjob.bat` batch file.

To start a job:

- Start a Command Prompt (cmd) on your LN server.
- Go to this folder `${BSE}\bin`.
- Run the `fillenv.bat` batch file to set the LN environment variables.
- Type `startjob.bat [job name]` to start the job.

Scheduling

To automate the execution of jobs, you can include startjob.bat in an external job scheduler, for example, the Windows scheduler. For this purpose, you must set the LN environment variables in the `startjob.bat` batch file. To achieve this, you can copy the settings in the `fillenv.bat` batch file to the `startjob.bat` file.

Run jobs for multiple companies

If you must start jobs that pertain to different company numbers, different versions of `startjob.bat` are created.

Chapter 14: BSE Services

A BSE Service is a UI-less 3GL session running in the background.

Actions are performed on behalf of other parts of the software. BSE Services must be implemented using a BSE Services Framework. All BSE Services share these common attributes:

- The service can be started and stopped manually on a specific application server. The service can be started automatically when the BSE becomes active.
- One instance per application server is started for the BSE.

BSE Services Parameters (ttaad8100m000)

The **BSE Services Parameters (ttaad8100m000)** session shows a check box indicating whether the BSE Services functionality is enabled. The **BSE Services Framework User**The service can be started automatically when the BSE becomes field is mandatory and you must specify a super user. The DB available timeout is default 45 seconds.

The bshell of an `ipc_info` entry to use is only visible on Windows. It must contain a value. The default value is: `bshell`.

When the BSE Services is activated:

- Standard BSE Services are added automatically, such as the Job Scheduler.
- Correct resources are written to `$BSE/lib/defaults/jobd`.
- Correct resources are written to `$BSE/lib/defaults/all`.
- The BSE Services Parameters table is ttaad800.

BSE Service Framework

BSE Services are implemented using a BSE Services Framework

BSE Services Framework Status (ttaad8101m000)

Running BSE Services are monitored. The **BSE Services Framework Status (ttaad8101m000)** session shows the status per host.

The BSE Service Framework status can be:

- **`Starting`**
- **`Started`**
- **`Stopping`**
- **`Stopped`**

- **Error**

If the BSE Services Framework cannot be started, the status is set to **Error**. For instructions to solve the issue, see the `$BSE/log/log.ttaad8210m000` log file.

BSE Services (ttaad8110m000)

Use this session to manage the BSE Services.

The administrator can add, configure, and start or stop services. The **BSE Services (ttaad8110m000)** session shows the service session code and description. With the **Automatic Start** check box you can specify whether the service must be started automatically.

BSE Services table – ttaad810

From this session you can start the **BSE Services by Application Server (ttaad8111m000)** session. Per host you can see the started BSE Service on a specific application server and the BSE Service status.

BSE Services by Host table – ttaad811

The BSE Service status can be:

- **Starting**
- **Started**
- **Stopping**
- **Stopped**
- **Error**

If a BSE Service terminates, either normal or abnormally, the status of this BSE Service is updated. BSE Services that terminate abnormally (crash), are restarted twice. If a BSE Service keeps terminating, the status of the BSE Service is updated to **Error**. For instructions to solve the issue, see the log file for the BSE Service: `$BSE/log/log.<BSE Service session code>`. The BSE Service must be restarted manually.

Windows jobd.exe service

If LN is deployed on a UNIX or Linux OS, the `jobd.exe` is replaced by a startup script in the `$BSE/etc` directory.

On Windows when the OS boots, the `jobd` service is started automatically. The `jobd.exe` is a porting set binary. The binary can start a bshell running any session the moment the application server becomes online. This is configured through the `startsession` and `stopsession` resources in `$BSE/lib/defaults/jobd`.

The **Activate Job Daemon (ttaad5206m000)** session is used by default as the start session.

The **Cancel Job Daemon (ttaad5207m000)** session is used by default as the stop session.

If `jobdbshell` is set, it points to an entry in `$BSE/lib/ipc_info, default $BSE/bin/ntb shell.exe`.

Starting and stopping BSE Services on Windows

On Windows, this process is started by the `jobd.exe` Windows Service when the application server becomes online.

The BSE Services tables are read and the specified BSE Services are started. These processes keep running until the bshell exits. The BSE Services by Application Server table is checked every five seconds to see if there are BSE Services to start. BSE Services with the **`Starting`** status are started. In this way a manual start of a BSE Service on a specific application server is supported.

No automatic restart of the bshell is done when the bshell stops unexpectedly.

Starting and stopping BSE Services on UNIX or Linux OS

If LN is deployed on a UNIX or Linux OS, you cannot use `jobd.exe`. Instead, you can use these scripts in the `$BSE/etc` directory to start and stop the services:

- `rc.start_bseservices`
- `rc.stop_bseservices`

To automate the process:

1. Include `rc.start_bseservices` in the `rc.start` script.
2. Include `rc.stop_bseservices` in the `rc.stop` script.

Job Scheduler

When BSE Services is enabled, the Job Scheduler is implemented as a BSE Service.

The Job Scheduler is a BSE service. When a job is queued, the Job Schedule data is added to the Scheduled Jobs (ttaad520) table. The Job Scheduler starts a job in a separate bshell. The job process reports whether it was successful.

Scheduled jobs are placed in a Scheduled Jobs table.

The ttaad520 table stores Job Schedule data per job and company, including status information.

The ttaad520 table contains these fields:

- **Job**
- **Company**
- **User**: the user for whom to run the job.
- **Next Execution Date**
- **Status**: **`Waiting`**, **`Starting`**, **`Running`**, **`Failed to Start`**
- **Hostname**: the Host on which the job is started
- **Bshell ID**: the PID of the bshell in which the started job runs

When the job is (temporarily) deactivated, the corresponding record in the Scheduled Jobs table is removed

Job Scheduler Service (ttaad5220m000)

The Scheduled Jobs table is checked every 60 seconds by the Job Scheduler Service. When a job must be started, the Job Scheduler Service starts this job in the correct company.

Started jobs run with the LN user specifics of who created the job. When the job process is started successfully, the scheduled job receives the **`Starting`** status.

If the job process cannot be started, the scheduled job receives the **`Failed to Start`** status. A new attempt to start the job is done in the next iteration

Job process

When the job process starts, it updates the Scheduled Jobs table. The job receives the **`Running`** status.

When the job process ends, it updates the Scheduled Jobs table:

- In case of a periodical job, the **Next Execution Date** is updated.
- In case of a non-periodical job, the scheduled job is removed.

Job data

All job management is done with the **Job Data (ttaad5500m000)** session. At some points the Scheduled Jobs table is updated:

- When a job is queued, a new record is inserted.
- When a job is canceled, the record is deleted.
- When a job is modified, the record is updated.

Activate Job (ttaad5203m000)

When started by the Job Scheduler Service, this session updates the scheduled job when the job starts. The bshell pid and the hostname are set.

When the job finishes, the session updates or deletes the scheduled job:

- In case of a periodical job, the new execution date is set.
- In case of a non-periodical job, the scheduled job is deleted.

Chapter 15: ES Service Manager

You can manage LN on your Windows server with an Infor ES Service Manager snap-in for the Microsoft Management Console (MMC).

This snap-in is displayed as the **Infor Manager** in the programs menu.

You can use the Infor ES Service Manager snap-in to run these processes:

- View all LN environments.
- View the status of the LN-specific services for each LN environment.
- Add, change, or remove LN-environment-specific variables.
- Install, start, and stop LN-specific services.
- Edit the settings of the LN-specific services.

Overview of the Enterprise Server Service Manager snap-in

The Enterprise Server Service Manager snap-in for the Microsoft Management Console (MMC) enables you to manage LN Software Environments (BSE) on your system.

The ES Service Manager snap-in installs automatically on the LN server during the installation of LN.

The ES Service Manager lists the LN environments (BSEs) on your system. You can independently administer the displayed BSEs.

The ES Service Manager displays these services:

- The ES Logic Service for every ERP LN installation.

  You can use the ES Service Manager to enable or disable the Logic Service for each LN installation.
- The Shared Memory Service for every BSE.

  You can use the ES Service Manager to enable or disable the Shared Memory Service for each BSE.

ES Logic Service

Use the ES Logic Service to configure the LN connection protocols. The ES Logic Service connects the LN Windows (BW) client to the LN server and checks whether the user has the authorization to connect to the LN server. With the ES Logic Service, you can start/stop access to the LN server.

To connect to the LN server, the ES Logic Service offers you these methods:

- Rexec

Also known as the remote execution protocol. `Rexec` does not use passwords encryption. We recommend that you disable this insecure protocol.

- BaanLogin

  Proprietary remote execution protocol. Sends user and password in encrypted form.

- Federation Services

  Provides central authentication and authorization for multiple applications. Federation Services makes single sign on (SSO) available for LN.

- Security Support Provider Interface (SSPI)

  This is only available on Windows.

Single sign on (SSO) permits a user access with a single action of user authentication and authorization. This is valid for all systems on which the user has access permission. You do not have to specify multiple passwords.

SSO does not use passwords, but uses the Windows user authentication. With SSO, users are no longer required to perform multiple logons, nor must users remember multiple passwords to access network resources. The user logs on to Windows once and no further passwords are required.

All SSO-specific information is stored in a single repository, the Active Directory, which provides a single, authoritative listing of each user's rights and privileges. You can change a user's privileges and the result is distributed network wide.

When using SSPI protocol for authentication, the Infor ES Logic service must run on a system, that can authenticate identities of incoming user requests. The Active Directory Server for the Windows Domain the LN server is member of is used/accessed. If that system cannot authenticate the user identity, the user logon is denied. Contact your local Windows administrator regarding trusts and authentication settings of the LN system and the Active Directory Server/Domain controller(s). Such relationship(s) must exist before (domain) users can be validated.

See the *Infor Enterprise Server Single Sign On User Guide*.

Shared Memory Service

With the Shared Memory Service, you can configure LN 's shared memory. The shared memory is a part of the physical memory intended for common use. Programs communicate with each other with shared memory. The use of shared memory results in faster access to the components loaded in shared memory. Before you start the application, you must initialize and set up the shared memory.

These components can be loaded into shared memory:

- Report objects
- Program objects
- Table definitions

LN management procedures

The ES Service Manager provides the tools to manage and administer LN on your system.

With the ES Service Manager, you have these options:

- Start or stop an LN Software Environment (BSE).
- Enable, disable, start, or stop the Shared Memory Service of the current BSE environment.
- Configure the ES Logic Service.
- Enable, disable, start, or stop the ES Logic Service for a system's LN environments.
- Configure the shared memory parameters.
- Rename BSE environments.
- Edit the BSE variables.

Note:

For ERP releases earlier than Infor Baan 5.2a, you can also use the Infor ES Service Manager to enable, disable, start, or stop these processes:

- The Job Daemon service by ERP installation.
- For porting sets earlier than 7.1d.12: the ERP Licensing Service by BSE. From porting set 7.1d.12 licensing is managed externally by the Infor Solution License Manager (SLM).

These options are not available for newer releases. For Infor Baan 5.2a and later, this applies:

- SLM manages licensing externally.
- For porting sets earlier than 8.3a the Application Services Manager (ASM) manages the Job Daemon service externally. From porting set 8.3 a you can choose whether you want to use ASM or the Job Daemon Service.

Note: Optionally, you can start, stop, and disable services through the Windows Service Manager, or through the -start, -stop, -install, and -remove command line options. We recommend that you perform all actions through the Infor ES Service Manager snap-in.

To start an LN software environment (BSE)

This procedure describes how to start or stop an LN software environment (BSE):

- Start the ES Service Manager from the Microsoft Windows Start menu.
- In the Tree pane of the ES Service Manager, expand the LN server node that contains the software environment.
- Select the LN software environment (BSE) you want to start and, on the toolbar, click the **Start Service** button. The BSE is now running.

To configure the ES Logic Service

To configure the ES Logic Service of an LN Software Environment (BSE):

- Start the ES Service Manager.
- Start the ES Logic Service Properties dialog box.
- Specify or modify the Service Properties
- Save the Service Properties changes
- Restart the Logic Service
- Restart the Shared Memory Service.

To configure the Shared Memory Service

To configure the Shared Memory Service of an LN Software Environment (BSE):

- Start the ES Service Manager.

- Select the LN Software Environment (BSE) for which you want to configure the shared memory.
- Start the **Shared Memory Properties** dialog.
- Specify or modify the shared memory properties.
- Save the changed shared memory properties.
- Restart the Shared Memory Service.

Note: The **Loaded Shared Memory** tab shows the loaded shared memory objects. The **Shared Memory Areas** tab shows a list of the available shared memory areas and their addresses.

To rename BSE environments

Ensure that the BSE environments on your system are unique. You can, therefore, rename environments if required.

To rename a BSE environment:

- Start the ES Service Manager.
- Rename a selected BSE environment.

To edit the software environment variables

You can use the Enterprise Server Service Manager to edit the software environment variables to change the behavior of the Enterprise Server.

To edit the software environment variables:

- Start the ES Service Manager.
- Start BSE Properties dialog box.
- Specify the variable name, for example, BSE_SORT.
- Specify the value of the variable. For example, for the BSE_SORT variable, specify the path to the directory where the temporary data must be stored during the sort process.
- Save the new variable.

Chapter 16: Language support

This section describes the language support capabilities of Infor Enterprise Server and the corresponding administrator tasks.

Character sets

A character set is a set of alphabetic or other characters used to construct the words and other elementary units of (a) native language(s).

During the installation of the LN application you must choose a character set. So only one character set applies for the whole LN environment. Therefore, only those languages can be stored which are supported by the character set that was chosen.

You can choose these character set types:

- single byte character sets
- multibyte character sets
- Unicode character set

Single byte character sets

Single byte character sets require one byte to store the character information. Therefore, max 256 characters are available. The ISO 8859 standard defines several characters sets, also called locales, to cover the characters of mainly the European languages.

Examples of single byte character sets are:

- ISO 8859-1: mainly west European languages like French, German, Italian etc
- ISO 8859-5: supporting Cyrillic languages (Russian).

The lower range, character 000 -127, is the same for all ISO 8859 character sets. The upper range character 128 – 255, is specific per locale.

The alphabet is encoded in the lower range and therefore the English language is supported with each ISO 8859 locale. English does not require any additional characters.

The sorting is binary based. The sorting is based on the order the characters are defined in the encoding. All uppercase alphabetical characters, for example, are sorted before the lowercase alphabetical characters. For example, the ‘Z’ is sorted before the ‘a’.

multibyte character sets

multibyte characters sets are typically required for languages that have more than 256 characters. A typical example is Chinese. In the context of LN the multibyte character sets require 4 bytes per character.

Examples of multibyte character sets are:

- BIG5: Chinese
- Wansung: Korean

Sorting is binary based.

Unicode character set

The Unicode character set is a standardized character set supporting all (modern) languages. This takes away the limitation of supporting a small set of languages within one LN environment. When you choose Unicode as character set, you can have for example Chinese, English and French in one LN environment.

Another advantage of the Unicode character set is that it comes with linguistic sorting rules. When the data must be visualized in a sorted form, the data is shown based on the sorting rules as defined by the ICU standard.

Note: The ICU standard also defines ‘tailoring’, that is fine-tuning the sorting rules to a specific language, which is not supported by the LN tools.

As a consequence, the database size of a Unicode based LN environment is bigger. The CPU and memory load on the system are higher than for a multibyte or single byte based character set. The choice for Unicode is typically made when multiple languages must be supported or when linguistic sorting is a preference.

High Ascii Tolerance

Note: This only applies to LN environments that do not run in Unicode mode.

You must set the high_ascii_tolerance resource to 0 in these situations:

- If your LN environment is a production environment and you plan to convert the environment to Unicode.
- If your LN environment is a development environment and you plan to deliver software components or translations. If high_ascii_tolerance is not set to 0, problems occur when the delivered components are imported in Unicode environments.

To set high_ascii_tolerance to 0, add this line in the $BSE/lib/defaults/all file:

```
high_ascii_tolerance:0
```

The role of the user locale

This section describes the role of the user locale in these types of installations:

- single-byte
- multibyte

- unicode

Caution: You technically can define a different locale for each user, see the **User Data Template (ttams1110m000)** session. This can cause issues. Therefore we do not support the usage of multiple user locales. Consequently all users in an LN environment must have the same user locale.

The role of the user locale in a single-byte installation

In a single-byte installation the user locale defines the character set that can be used throughout the application.

Caution:
- Use a user locale that matches the character set used in the database. In this way “3rd party” database tools can access the data.
- Use a locale that defines a binary sorting order, also known as collation.
- Choose a binary sorting order in the database as well. For example, in West-European countries you can use the ISO_BIN1 locale. The character set of this locale is the same as the ISO-8859-1 locale, but the sorting order is binary.
- Ensure all users have the same user locale. Note that this is not enforced by the tools or porting set.

The user locale has affect on these processes:

- The way data is stored in the database. The user locale determines the code points that are used to store single-byte characters.
- Some bshell (3GL) functions:
  - mb.locale.info(), including the TSS_GET_IFACTOR and TSS_GET_EFACTOR aspects
  - set.min()
  - set.max().
  - set.fmin()
  - set.fmax()
- The non-Unicode version of BWPrint, which must convert to the correct Windows code page. Note: to convert data from the ISO locales to Windows code pages, BWPrint uses the _WIN32 versions of the user locale.

The role of the user locale in a multibyte installation

In a multibyte installation the user locale defines the character set that can be used throughout the application.

Caution:
- Use a user locale that matches the character set used in the database. This character set must use a binary sorting.
- Data is stored in the database using the character set of the user locale. Ensure the correct character set is specified for the database, so the database treats the characters in the correct way. In this way “3rd party” database tools can access the data. Otherwise the data could show up garbled.
- Ensure all users have the same user locale. Note that this is not enforced by the tools or porting set. If users have different user locales, conversion errors occur when a user processes data of

another user, who has a different user locale. This mainly impacts processes that write to and read from the database. Consequently the user locale impacts any integration that interacts with the database.

The user locale has affect on these processes:

- The conversion of data from “native” format (MBCS) to TSS and vice versa. The user locale determines the meaning of the term “native”.
- The way data is stored in the database. The user locale determines the code points that are used to store multibyte characters.
- Some bshell (3GL) functions:
  - mb.import$(). This function converts data from “native” format to TSS.
  - mb.export$(). This function converts data from TSS to “native”.
  - utf8.export$(). The Baan IV porting set uses the “native” format as intermediate format to convert from TSS to UTF-8. If the user locale does not match with the data, this can result in conversion errors.
  - mb.width(). This function returns the width of a string, where width is defined in “number of display positions”. For example, in the ISO8859n character sets, the `LATIN SMALL LETTER E WITH ACUTE` character takes one display position. In the `GB2312` character set, it takes two display positions.
  - mb.locale.info(), including the `TSS_GET_IFACTOR` and `TSS_GET_EFACTOR` aspects.
  - set.min()
  - set.max()
  - set.fmin()
  - set.fmax()
- The appearance of log messages. The text in log messages that is converted from the TSS character set to the “native” format.
- The non-Unicode version of BWPrint, which must convert TSS data to “native” format.

The role of the user locale in a Unicode installation

Because the introduction of the Unicode character set, the role of the user locale has become less important. In a pure Unicode environment all characters are represented by unique code points. All code points have a unique interpretation. There are still some areas where conversions from and to Unicode occur.

Example

You work in a Unicode environment. But your personal user locale is ISO8859. You want to exchange data between the Unicode environment and another environment. When you perform an export from the Unicode environment, for example through LN Data Director or EDI, the export files are in ISO8859 format.

The user locale has no affect on these processes:

- The way data is stored in the database. All data in multibyte columns is stored in Unicode. The data in single-byte columns is stored “as is” in Unicode. Effectively it is interpreted in the ISO-8859-1 locale. For example, the `LATIN LETTER A` is stored as Unicode code point 0x41. The code point

0x9e is stored as the Unicode code point 0x9e, which represents the `LATIN SMALL LETTER E WITH ACUTE` character (é).

- The normal operation of the bshell. This excludes these functions:
  - Conversion functions, such as mb.export$() and mb.import$().
  - Functions to acquire information about the current user locale, such as mb.locale.info().

  These functions are not affected by the user locale:
  - set.min()
  - set.max()
  - set.fmin()
  - set.fmax()
- The appearance of log messages. The text that is in log messages is converted from UTF-T to UTF-8 format.
- The Unicode version of BWprint.

The user locale has a small affect on the dump files as created by the `bdbpre` utility. Data in the `bdbpre-dump` files is in the UTF-8 character set. If the database contains “high ascii” characters, these characters are converted in the context of the current user locale. Note that the high_ascii_tolerance resource has no effect on this process. See the comment on the conversion of “high ascii” characters.

The user locale has affect on these processes:

- The conversion of data from “native” format (MBCS) to Unicode (UTF-T) and vice versa. The user locale determines the meaning of the term “native”.
- The conversion of so-called “high ascii” characters. See below.
- Some bshell (3GL) functions:
  - mb.import$(). This function converts data from “native” format to UTF-T.
  - mb.export$(). This function converts data from UTF-T to “native”.
  - mb.width() This function returns the width of a string, where width is defined in “number of display positions”. For example, in the ISO8859n character sets, the `LATIN SMALL LETTER E WITH ACUTE` character takes one display position. In the GB2312 character set, it takes two display positions.
  - mb.locale.info(), excluding the `TSS_GET_IFACTOR` and `TSS_GET_EFACTOR` aspects.
- The non-Unicode version of BWPrint, which must convert UTF-T data to “native” format. This can result in conversion errors because the “native” character set supports only a limited subset of UTF-T. Therefore, we recommend that you use the Unicode version of BWPrint in a Unicode installation.

Conversion of “high ascii” characters

The occurrence of “high ascii” characters poses a problem, because one code point can have different meanings in different character sets.

Example

In the ISO-8859-1 locale, the code point 0xe9, decimal 233, is interpreted as the `LATIN SMALL LETTER E WITH ACUTE` character (é).

In the ISO-8859-7 locale, this code point is interpreted as the `GREEK SMALL LETTER IOTA` character (ι).

To determine the meaning of a “high ascii” character, LN uses the current user locale. If the user locale is an ISO8859n variant, then this character set is used to determine the correct meaning. Otherwise the ISO85591 character set is used.

Example

The user locale is ISO88597. A string, which contains the 0xe9 code point, must be converted to UTF-T. The code point is interpreted as the `GREEK SMALL LETTER IOTA` character. The resulting UTF-T code point is 0x9bbc87b9.

The user locale is ISO88591 and the same string must be converted. The code point is interpreted as the `LATIN SMALL LETTER E WITH ACUTE` character. The resulting UTF-T code point is 0x9bbc81e9.

We recommend that you keep the installation clean from “high ascii” characters. To achieve this, set the high_ascii_tolerance resource to 0.

Conversions

Character sets can have several encodings. A certain character is represented by an integer and that integer is converted to a certain encoding.

Unicode for example has these standard encodings:

- UTF-8
- UTF-16
- UTF-32

For LN a Unicode encoding is introduced that fits most with the current architecture. This encoding is called UTF-T and is in line with the TSS concept. TSS is the LN specific encoding for the other character sets.

UTF-T is only used during runtime, that means that storage of data is always done in a standard encoding.

UI

Worktop/BW/BWPrint

These clients perform a conversion from Windows Unicode encoding to UTF-T or another TSS character set. That depends on the character set of the LN environment. Communication between these clients and the server is in TSS format.

Web UI

Web UI handles all data in Unicode and passes the data to the server in standard UTF8. On the server the transformation to the TSS character set, in case of Unicode to UTF-T, takes place.

Reporting

- On UNIX

  On UNIX the support for printing in various languages is limited. The printer drivers, in contrast to Windows, provide limited conversion support. Therefore the capabilities of the printer determine which languages can be supported.

- On Windows

  On Windows the LN reports are printed through BWPrint. BWPrint converts to the Windows Unicode encoding and therefore any language can be printed.

Integration by the Adapter for LN

The Adapter for LN communication is based on Unicode.

Language Translation Support

LN uses Language Translation Support (LTS) to reduce the language dependency of the LN applications and lower the costs of media creation and distribution.

For details, see:

- Language Translation Support in the Infor Web Help.
- To translate software components in the *Infor LN Development Tools Development Guide*.

Chapter 17: Multilanguage application data

If your LN system uses the Unicode character set, you can run multiple languages of the LN software, such as Chinese, English, and French in one LN environment.

You can enable LN to store shared application data in multiple languages. This is useful if users from all over the globe are using the same environment. Depending on the user's software language, the application data can also be displayed in the same language. If a report for a business partner is printed, the layout and data of the report are in the business partner's language.

To store shared application data in multiple languages, you must configure Multilanguage Fields Support.

Note:

You can only configure Multilanguage Fields Support for fields of the Multibyte String data type.

To support all languages available for LN in one environment, you must install LN in Unicode mode. Unicode is the standard for multilanguage support and provides a unique code for each character regardless of the language.

ISO standards

In Infor Enterprise Server versions earlier than 8.4.1, the coding of data languages is based on the ISO 639-2 standard. This standard is not sufficient for coding languages such as Simplified Chinese, Traditional Chinese and Portuguese, and Brazilian Portuguese.

To support these languages, language coding in Infor Enterprise Server 8.4.1 and later is done according to the ISO standards used for Java programs.

A data language code consists of:

- A language code based on the ISO 639-1 standard.
- Optional: a country code based on the ISO 3166-1 standard.

The language code and the country code are separated by an underscore (_), see this table:

Data Language	Description
de	German
en_GB	English_UNITED KINGDOM
en_US	English_UNITED STATES
he	Hebrew
it	Italian

Data Language	Description
ja	Japanese
nl	Dutch; Flemish
pt_BR	Portuguese_BRAZIL
pt_PT	Portuguese_PORTUGAL
zh_CN	Chinese_CHINA (Simplified)
zh_TW	Chinese_TAIWAN (Traditional)

Note: To use the new Enterprise Server 8.4.1 coding standards, a table reconfiguration can be required for data languages created in earlier Enterprise Server versions. For details, see the Specific Installation Guide - Updates (U9497).

Language types

This table shows the language types:

Language type	Description
Software Language	The language in which the software components, such as forms and reports, are displayed. For each user, the default software language is defined in the LN user data.
Data Language	The language in which the application data is displayed. Data languages are linked to users and software languages.
Base Language	The fallback language used to display application data. LN uses this fallback mechanism: **1** LN displays the application data in the data language specified in your user data. **2** If no data language is specified in your user data, LN displays the application data in the data language linked to your software language. **3** If no data language is linked to your software language, LN displays the application data in the base language.

Configuration Multilanguage Fields Support

To support multilanguage fields, LN must be installed in Unicode mode. You can choose the Unicode character set in the LN Installation Wizard.

For more information on the prerequisites for a Unicode installation, see the Technical Notes.

Before you start the configuration of Multilanguage Fields Support, it is recommended you create a backup of the application data. For example, you can extract your application data through the **Create Sequential Dump of Table (ttaad4226m000)** session, or you can create a backup through the backup tools in your RDBMS.

1 Enable support for Multilanguage Fields.

a Start the **Parameters Multi Language Support (ttadv4106m000)** session.

b Configure Multibyte and Unicode, and perform the required steps for this configuration. See the instructions displayed in the session's window.

c Select the **Multi Language Fields Support** check box.

d Use the **All Data Languages Always Available** check box to specify whether all language variants must be read:

- If this check box is selected, then for multibyte fields that are “Multi Data Language Enabled” all language variants are read.

  This might result in reduced performance in sessions with multilanguage fields.

- If this check box is cleared, then only the language variants of the user's language are read. When a report is printed, the application can select the language that is read from the database or force to read all languages.

e Save the changes and close the session.

2 Define data languages.

a Start the **Data Languages (ttaad1111m000)** session.

b Define the desired data languages. A data language consists of an ISO 639-1 language code and, optionally, an ISO 3166-1 country code. You can zoom to the **ISO 639 Languages (ttaad1512m000)** session and the **ISO 3166-1 Countries (ttaad1513m000)** session.

c Define the base language: Select a data language and, on the *appropriate* menu, click **Make Base Language**.

The base language is used as a fall back language. Application data is displayed in the base language if both of these scenarios occur:

- No data language is defined in the user data of an LN user.
- No data language is linked to the user's software language.

d Convert the data languages to the runtime data dictionary. On the *appropriate* menu, click **Convert Data Languages to Runtime**. The conversion process generates the `$BSE/lib/data_langs` file.

Note:

- Only perform this step if you are sure you no longer want to change the base language. See the restrictions listed in the online help of the Data Languages (ttaad1111m000) session.
- During the conversion to runtime, all users must be logged off.
- When the conversion to runtime has finished, you must restart the LN environment.

e Close the session.

Data Languages have a status that is used to control their life cycle.

See "Life cycle of a data language" in the online help of the Data Languages (ttaad1111m000) session.

3 Link data languages to the corresponding software languages.

a Start the **Software Languages (ttaad1510m000)** session.

b Double-click a software language, to start the **Software Languages (ttaad1110s000)** session. In the **Data Language** field, specify the desired data language.

Note:

- You can zoom to the **Data Languages (ttaad1111m000)** session.
- You can only select data languages with status "Base Language" and "Available".
- If the **Data Language** field is blank, the application data is shown in the data language that is defined in the LN user data. Or in the base language.

c Save the changes and close the sessions.

To ensure that application software and application data are displayed in the same language, link the data languages to the software languages.

4 Register Multilanguage Tables and Table Fields that must be multilanguage enabled.

a Start the **Registered Tables with Multi Language Fields (ttadv4137m000)** session.

b Select a package combination.

c Add the tables to be Multilanguage Enabled.

d Per table, specify whether all languages must be read for records from the table:

- If all languages must be read, select the **All Data Languages Always Available** check box.
- Otherwise, clear this check box.

Note:

- If all data languages are always available, this can result in reduced performance in sessions with multilanguage fields.
- You can only clear this check box if the application feature pack is high enough.
- The **All Data Languages Always Available** setting is saved in the data dictionary of the table. Therefore, after a change in this setting, a conversion to runtime for tables is required.

e Double-click a table. The **Registered Multi Language Fields (ttadv4138m000)** session starts.

f For each field to be multilanguage enabled, select the **Enabled** check box.

g Save the changes and close the sessions.

5 Convert the multilanguage settings to the runtime data dictionary.

a Start the **Convert to Runtime Data Dictionary (ttadv5215m000)** session.

b Select the **Tables** check box and the **Reconfigure Tables** check box.

c For package combinations, packages, and tables, specify the appropriate selection ranges.

d Click **Convert to Runtime**.

e Optionally, start the **User Data (ttaad2500m000)** session and define a data language per user.

The data language that is specified in the user data takes precedence over the data language that is linked to a software language.

See "Language types" in Multilanguage application data on page 159.

Note: If you use the Enterprise Modeler Content Pack with LN, consider using the MAA0050 (Multi Data Languages for Descriptions) wizard to set up multilanguage support. You can execute this predefined wizard from the **Wizards by Project Model (tgwzr4502m000)** session after you specified the business function model for your company.

Data translation

When a user inserts new records, the multilanguage field descriptions are stored in the user's data language.

LN automatically copies the new field descriptions to the other data languages. These descriptions must be translated. This is to ensure that all users who are running another data language code can read the new fields in their own language.

We recommend to assign dedicated users to perform the data translation; these users must be authorized to switch between different data languages.

Ensure that a user can switch to another data language. Select the **Allow Changing of Data Languages** check box in the user's user data template properties. See the **User Data Template (ttams1110m000)** session.

Web UI and Worktop users can switch to another data language.

This table shows where to find this option in the UI:

Web UI user	Select **Options > Infor ERP Options > Change Data Language**
Worktop user	Sellect **Tools > Change Data Language**.

Insertion and translation process

The tasks to translate multilanguage field descriptions depend on the data language the new records are specified in. See these sample scenarios:

Scenario 1: A user works in the base language and inserts a record

1. The user inserts a record. The descriptions of the multilanguage-enabled table fields are inserted in the base language code, and are automatically copied to all other linked data languages.
2. For each table that is involved, you must run the **Print Multi Language Application Data (ttadv4438m000)** session. The session prints a report that shows the records that must be translated to other languages.
3. Switch to the other data language(s) and translate the descriptions of these records.

Scenario 2: A user works in another language and inserts a record

1. The user inserts a record. The descriptions of the multilanguage enabled table fields are inserted in the other language code. They are automatically copied to all other available data languages and the base language.
2. For each table that is involved, you must run the **Print Multi Language Application Data (ttadv4438m000)** session. The session prints a report that shows the records that must be translated.
3. Switch to the base language and translate the descriptions of these records to the base language. The translated descriptions are automatically copied from the base language code to all other available data language codes. Except the language code in which the records were created first.

4 Run the **Print Multi Language Application Data (ttadv4438m000)** session again for each table that is involved. The session prints a report that shows the records that must be translated from the base language to other languages.

5 Switch to the other data language(s) and translate the descriptions of these records.

Check whether the descriptions that are stored in the original language code are still correct. That is, the language code in which the records were created.

To remove the corresponding records in the check report, save a dummy change of these descriptions.

Chapter 18: Sensitivity labeling

Sensitivity labeling enables you to provide feedback to the end user about the sensitivity of the information on an LN screen.

Sensitivity labels have an inherent hierarchy, as each label implies a more restrictive sensitivity than the next. For example, sensitivity labels have different levels of sensitivity, such as “Company Sensitive” and “Highly Sensitive”, which increase in order of sensitivity.

You must define the sensitivity level at either the table-field level or at the table level. Each field or table requiring a certain sensitivity level must be assigned the appropriate sensitivity label. When you specify a table, all fields in the table have the same sensitivity level, except those fields for which you have defined another sensitivity.

You can assign a sensitivity label to individual sessions and reports. Sometimes the assignment that is based on table fields is not applicable. For example, a calculated value on a report may be sensitive because it is based on a sensitive table field.

Sensitivity labels are shown on forms and reports.

Configuring sensitivity labeling

1 Enable sensitivity labeling

Start the **Tools Parameters (ttaad0100m000)** session and select the **Sensitivity Labeling enabled** check box.

2 Define sensitivity labels

Start the **Sensitivity Labels (ttaad3150m000)** session.

Use this session to perform these actions:

- Define sensitivity labels and their sensitivity level.
- Assign color schemes, which are only used during form presentation, to sensitivity labels.

3 Assign sensitivity labels to table fields

Start the **Sensitivity Level of Table fields (ttaad3151m000)** session. You can use this session to assign sensitivity labels to tables or table fields.

With the **Print Impacted Components (ttaad3450m000)** session you can print a list of reports that gets a sensitivity label based on the defined fields.

4 Optionally: Assign sensitivity levels to specific sessions and reports

You can use the **Sensitivity Level of Reports and Sessions (ttaad3152m000)** session to assign sensitivity levels to specific sessions and reports.

5 Convert sensitivity labels to runtime

Convert the sensitivity label definitions to the runtime data dictionary. This generates various files, whose names start with ‘sensitivity’ in the `$BSE/lib` directory.

At runtime, LN uses these files to identify whether sensitivity labels must be applied.

Runtime behavior

- Sessions

  LN determines the sensitivity label on session level. If multiple fields of a session have sensitivity labels assigned, the most sensitive label is allocated.

  The label is displayed in the upper left corner of the data area of the form. The font size is a predefined size of 1.5, which is the default size, and the font style is bold.

  The color of the label is based on the color scheme assigned during configuration.

- Reports

  The sensitivity label is repeated and displayed in the upper left corner of each new page.

  Output devices with font size capabilities, such as PDF and Windows printing, use these font settings:

  - Font size: 1.5 times the default font of the report.
  - Font style: bold.

6 Run a conversion to runtime for users

You must run a convert to runtime for users in the **User Data (ttaad2500m000)** session when the tools parameter **Sensitivity Labeling enabled** check box in **Tools Parameters (ttaad0100m000)** is changed.

Chapter 19: Shared Memory

The shared memory is a part of physical memory intended for common use. Programs communicate with each other through shared memory.

Shared memory results in faster access to the components loaded in shared memory. To enhance the performance of your LN system, you can load program objects and report objects into the shared memory. Before you set up and initialize shared memory, you should verify that the hardware and internal memory is sufficient.

After the installation of LN, various objects are already loaded in the shared memory. This section describes the procedure to load additional objects into the shared memory.

LN uses many DLLs; therefore, it is difficult to decide which objects should be loaded into the shared memory. Before you fill the shared memory, use measurement tools to identify which objects are frequently used and suitable to load into the shared memory.

A shared memory that is configured based on the measurement information improves the processing speed of your LN application.

For more information on shared memory, see *Infor Enterprise Server Technical Manual*.

1 Activate logging

To gather statistics on the usage of objects, start the Logging Parameters (ttaad4153m000) session.

Optionally, click **Cleanup** to remove the currently stored shared memory measurement data.

Specify the time frame in which shared memory logging must take place. If you do not define a time frame, no logging is done.

After you specify the logging parameters, click **Convert to Runtime**. You can only run the **Convert to Runtime** command on a Master Application Server.

The measurement starts for all bshells started after the conversion is run, taking the defined time frame into account. The logged information of a bshell is stored when the bshell is closed.

For details, see the session help.

2 View object usage statistics

After the period of shared memory measurement has ended, start the Shared Memory Object Measurement (ttaad4152m000) session.

Use this session to view statistics on the object usage in the bshell. The session displays, among other things, the number of times an object was loaded during the measurement period. This information helps identify which objects are frequently used and should be loaded into the shared memory.

Optionally, click **Print** to print a Shared Memory Object Measurement report.

Optionally, on the *appropriate* menu, click **Cleanup** to remove the shared memory measurement data.

For details, see the session help.

3 Optional: Allocate shared memory for package combinations.

Start the Shared Memory Data (ttaad4156m000) session and click **Allocate**. The Allocated Memory Shared Memory (ttaad4155m000) session starts. Specify, per package combination, the amount of memory, in MBs, that must be allocated for object usage in shared memory.

Note: The amount of shared memory for the package combination is unlimited if you do not allocate memory for a package combination. Or if you allocate 0 MB for a package combination.

4 Specify shared memory data

Start the Shared Memory Data (ttaad4156m000) session. In this session, you can specify shared memory data in these ways.

Import objects from measurement (Recommended)	On the *appropriate* menu, click **Import Objects from Measurement**. The Import Objects from Measurement (ttaad4255m000) session starts. Use this session to: • Import a range of program objects or report objects from the object measurements into the shared memory data. • Import the most frequently used objects until the allocated memory, defined in the Allocated Memory Shared Memory (ttaad4155m000) session, is reached. To do this, you must select the **Fill remaining allocated memory with most counted objects** check box. The current shared memory data is deleted first, except the objects marked as Fixed. This option is recommended. A shared memory that is filled with the most frequently used objects improves the processing speed of your LN application.
Import objects from DD	On the *appropriate* menu, click **Import Objects from DD**. The Import Objects from DD (ttaad4254m000) session starts. Use this session to import a range of program objects or report objects from the Data Dictionary (DD) into the shared memory data.
Manually add objects	To manually enter objects in the grid, click **New**.

Note:

- You can define one or more objects for a package combination. The domains and table definitions of that package combination are also loaded into the shared memory. You cannot add individual domains and table definitions.
- Tools objects, objects of packages tt and tl, do not belong to a package combination, and are therefore stored in the dummy <empty> package combination. To create a list of Tools objects to be loaded into shared memory, leave the **Package Combination** field empty.
- You can mark objects as Fixed. Fixed objects are not removed from the shared memory data when running the Import Objects from Measurement (ttaad4255m000) session with the **Fill remaining allocated memory with most counted objects** check box selected.

For details, see the session help.

5 Convert the shared memory data to Runtime

On the *appropriate* menu in the Shared Memory Data (ttaad4156m000) session, click **Convert to Runtime**. You can only run this command on a host that is defined as Master Application Server (MAS).

When the conversion is complete, click **OK** and close the session.

6 Restart the shared memory

The procedure to restart the shared memory depends on the operating system of the LN server.

Operating System	Procedure
Windows	To restart the shared memory: **a** Start the Enterprise Server Service Manager. **b** Right-click the Shared Memory Service, and on the shortcut menu, click **Stop**. **c** Right-click the Shared Memory Service again, and on the shortcut menu, click **Start**.

UNIX	To restart the shared memory:
	a Log off from LN.
	b Log onto the LN server as user root.
	c Specify these commands:
	`# cd $BSE/bin`
	`# shmmnanager6.2 -k`
	`# shmmnanager6.2 -i`
	`# ./srdd_init6.2 -i`

Chapter 20: Customer defined fields

You can use the Customer Defined Fields (CDF) sessions in LN to store additional information in tables without creating so-called customizations. These sessions are available in the **Tools** menu, under **Application Configuration**.

Limitations

- You cannot define customer defined fields for tables within the tl and tt Tools packages.
- External integrations, such as Infor Integration, EDI, Office Integration, and SOA-based integration, do not support customer defined fields.
- You can use customer defined fields within 4GL reports, as far as editing the 4GL report layouts is still supported in your environment. For external reporting, only Infor Reporting and Microsoft Reporting (SSRS) support customer defined fields.
- There is no direct limitation on the number of CDFs in a table. The actual number of fields in a table and the total length of all fields may be limited by the RDBMS you use.
- Only super users can run the **Convert to Runtime Data Dictionary (ttadv5215m000)** session to convert the customer defined fields and the related domains to the runtime data dictionary.

Note: The full functionality of customer defined fields is only available within Web UI and LN UI. Customer defined fields are not displayed in the classic LN BW UI.

Personalization

A session can contain two types of customer defined fields:

- Customer defined fields of the main table. These are regular customer defined fields. If set to active, these fields are initially visible in the session. See Customer Defined Fields (ttadv4591m000) .
- Referenced customer defined fields. These are active customer defined fields that belong to a table that is referenced by the main table of the session. Referenced customer defined fields are initially invisible in the session.

A referenced customer defined field is linked to a session if its table is referenced through:

- A table definition (so-called hard-reference)
- A reference specified in the form definition, in particular in the **Input/Ref Express** property of a form field
- A `query.extend.select()` call in the UI script of the session

If a session contains customer defined fields, you can perform these actions:

- Hide/unhide these fields.
- Move these fields to a different location, such as a different tab.

See Personalization Workbench (ttstppers).

Related documentation

For more information about customer defined fields, see "Customer Defined Fields" in the *Infor LN Extensions Development Guide*.

Configuring customer defined fields

1. Activate the CDF feature.
   a. Start the **Customer Defined Fields Parameters (ttadv4590m000)** session.
   b. Select **CDF Active** and click **OK**.
2. Define customer defined fields.

   Define customer defined fields in one of these ways:
   - Use the **Customer Defined Fields** option in the **View** menu of an already started session.
   - Use the **Customer Defined Fields (ttadv4591m000)** session.

     If you want to create customer defined fields of type ‘List’, specify the lists and their constants in the **Lists (ttadv4592m000)** and **List Constants (ttadv4593m000)** sessions.
3. Convert to runtime.
   a. Start the **Convert to Runtime Data Dictionary (ttadv5215m000)** session.
   b. Convert the customer defined fields and the related domains to the runtime data dictionary.

Note: If you use the Enterprise Modeler Content Pack with LN, consider using the MIT0200 (Customer Defined Fields) *wizard* to set up Customer Defined Fields. You can execute this predefined wizard from the **Wizards by Project Model (tgwzr4502m000)** session after you specified the *business function model* for your company.

Chapter 21: Scrollbars

Several scrollbars are supported by the LN sessions.

This table shows the types of scrollbars that are supported:

Old scrollbar	This scrollbar is inaccurate. The scrollbar slider has a fixed size. It does not reflect the number of records in the session. The slider is at the top, in the middle, or at the bottom of the scrollbar. It does not indicate the relative position in the dataset. This scrollbar is only supported in LN UI and Worktop. In Web UI, navigation buttons are displayed instead of this scrollbar.
New scrollbar	This scrollbar is more advanced. The size of the scrollbar slider is inversely proportional to the number of records in the session. The position of the slider indicates the relative position in the dataset. This scrollbar is supported in LN UI, Worktop, and Web UI.

The new scrollbar requires calculation at session startup. Therefore it can affect the performance of a session. The impact on the performance is proportional to the number of records in a session.

To avoid a performance decrease:

- Disable the new scrollbar for sessions that contain many records.
- Disable the new scrollbar for any session, irrespective of the number of records. This is useful for sessions with not so many records, which start slowly.

If you start a session with a disabled scrollbar in LN UI or Worktop, the old scrollbar is displayed.

If you start the session in Web UI, no scrollbar is displayed. Instead, navigation buttons are displayed.

Disabling the scrollbar for sessions with many records

To disable the scrollbar for sessions with many records:

1. Start the **Tools Parameters (ttaad0100m000)** session.
2. In the **Runtime Count Limit** field, specify a value.

   The scrollbar is automatically disabled for sessions that contain more records than the specified value.
3. Save the changes and close the **Tools Parameters (ttaad0100m000)** session.

Disabling the scrollbar for any session

To disable the scrollbar for a session:

1. Start the **Tools Parameters (ttaad0100m000)** session.
2. Next to the **Runtime Count Limit** field, click **Sessions**.
   The **Sessions with a Runtime Count Limit (ttaad4180m000)** session starts.
3. Add the session to the list, for which to disable the new scrollbar.
   See the session help.
4. Save the changes and close the sessions.

Chapter 22: Image parameters

In some LN sessions, Web UI users can drag an image from a web browser or Windows explorer window, and drop it on the session's form. The dropped image is stored in an image repository on the server.

Each image in the image repository belongs to an image set. Each image set is linked to an LN table in a particular company number.

The image repository consists of these components:

- Images (ttaad710) table

  Do not share this table between companies.
- An image folder

  By default, this folder is located in `${BSE}/images`. You can change the location of the image folder.

The location of the image folder is stored in the Image parameters (ttaad700) table. To view images in sessions and to drop images on forms, users require authorizations.

Note: Dragging and dropping images is only supported in sessions where the image is linked to the current main table or secondary table.

Changing the location of the image folder

1. Ensure all users have signed out from LN.
2. Start the **Image Parameters (ttaad7100m000)** session.
3. In the **Target Directory** field, enter the new directory.
4. Click **Move Directory**.
5. Sign out from LN and sign in again.
6. If required, modify the user authorizations, see Authorizations on page 175..

Note: To view the contents of the image folder, in the **Image Parameters (ttaad7100m000)** session, click **Images**.

Authorizations

To view images in sessions, users require these authorizations:

- Read authorization, on operating system level, on the image folder and its contents.
- Read authorization on the Images (ttaad710) table, see Table authorizations on page 81.

To drop images on forms, users require these authorizations:

- Read and Write authorization, on operating system level, on the image folder and its contents.
- Read and Write authorization on the Images (ttaad710) table.

Chapter 23: HTTPS

HTTPS, (Hypertext Transfer Protocol Secure), is not a separate protocol. It refers to the combination of a normal HTTP interaction over an encrypted Secure Sockets Layer (SSL) or Transport Layer Security (TLS) connection.

HTTPS connections are often used for payment transactions on the World Wide Web and other sensitive transactions.

HTTPS support with SOAP

Simple Object Access Protocol (SOAP), is a standard protocol that is used to communicate with Web Services.

To use HTTPS with SOAP, some prerequisites must be met. The correct versions of the porting set, 4GL Engine and Application Studio must be installed. The required certificates that enable HTTPS must be installed and accessible by the porting set.

Prerequisites

To use HTTPS with SOAP meet these prerequisites:

- Porting set 8.7a.02
  - Porting set 8.7a.02 is the first porting set to contain correct support for HTTPS and SSL. The porting set uses the cURL and OpenSSL libraries, and offers 3GL functions to enable HTTPS.
- 4GL Engine build 3971, solution 1009866
  - The SOAP DLL in this build of the 4GL Engine contains functions such as SoapService.SetClientCertFile() and SoapService.SetCertAuthorityDir() which are required for HTTPS.
- Application Studio build 8.7.0.0229
  - The WSDL2Baan tool in this build of the Application Studio supports HTTPS and generates wrapper functions to create client and server certificates.
- OpenSSL tool version 1.0 (recommended)
  - The OpenSSL command line tool is required for converting certificates from DER and PKCS12 format to PEM format. It can be used to generate the hashed file names of server certificates, see "To generate the hashed file name of a certificate". It can be used to diagnose issues connecting to web sites. If the correct certificates are already available in PEM format, and no hashed file names are required, the OpenSSL tool is not required.

- Perl (optional)
  - Perl (Practical Extraction and Reporting Language) is a scripting language. It is required for the c_rehash utility script, which may be used to create hashed file names of server certificates. Note that this utility is not shipped with every distribution of OpenSSL.

Certificates to enable HTTPS

We recommend that certificates are stored in the `$BSE/security/certs` folder, under the client and server sub-folders.

The administrator must install the appropriate certificates in the correct location. Guidance on how to obtain the certificates, see Dealing with certificates on page 178.

Client certificates

Client certificates authenticate the client side of a connection to the server side (website or web service). Client certificates can be used to authenticate a user, an organizational unit, or a server. The client certificate is presented to the server during the SSL negotiation. If the server cannot verify the client certificate, the connection is refused.

For security reasons the client certificate must be password protected. The password must be kept secret. Anyone having access to both the certificate and the password can assume the identity of the legitimate owner of the certificate.

Preferably, client certificates must be stored in PEM format in the `$BSE/security/certs/client` folder.

The client certificate is specified with the `SoapService.SetClientCertFile()` SOAP function. Its accompanying password is set with the `SoapService.SetPrivateKeyPasswd()` SOAP function.

Server certificates

So-called server certificates authenticate a server. They can be self-signed (less secure) or signed by a recognized certificate authority (CA) such as VeriSign. The server certificate is presented to the client during the SSL negotiation. The client trusts the server if it has access to the root certificate of the server certificate. If the server certificate cannot be verified, the connection is refused.

Preferably, server certificates must be stored in PEM format in the `$BSE/security/certs/server` folder. The administrator can decide to store the certificates in another location.

If the `SoapService.SetCertAuthorityDir()` SOAP function is used, the hashed file names must exist in that directory, and link to or have the contents of the appropriate certificate. An example of a hashed file name is 415660c1.0, which is a hashed file name of the `vsign3.pem` certificate.

You can also use a so-called CA-bundle, which is a file that contains multiple (root) certificates. In that case the `SoapService.SetCertAuthorityFile()` SOAP function must point OpenSSL to the file name of the CA-bundle.

Dealing with certificates

Certificates must be in PEM format and can require hashing for OpenSSL to find the correct certificate.

To find out which server certificate is required

To find out which server root certificate is required by a secured website or web service, use the OpenSSL command `openssl s_client –showcerts –connect host:port`. For example:

```
openssl s_client -showcerts -connect ccapi.client.qvalent.com:443
```

In the output of the command you find the certificate chain that is used by the secured connection. Look for the words “Certificate chain”. Under these words you find text that resembles this:

```
0 s:/C=AU/ST=NSW/L=Wallsend/O=QVALENT Pty Ltd/OU=Terms of use at
www.verisign.com/rpa (c)00/CN=ccapi.client.qvalent.com i:/O=VeriSign Trust
Network/OU=VeriSign, Inc./OU=VeriSign International Server CA - Class
3/OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

1 s:/O=VeriSign Trust Network/OU=VeriSign, Inc./OU=VeriSign International
Server CA - Class 3/OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY
LTD.(c)97 VeriSign i:/C=US/O=VeriSign, Inc./OU=Class 3 Public Primary Certi
fication Authority

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE----- ---
```

The issuer of a certificate, preceded with “i:” in the output, is the subject of the certificate, preceded with “s:” in the output that is next in the chain.

The required root certificate is the one whose subject equals the issuer of the last certificate in the chain.

Converting several formats to PEM format

The OpenSSL command can be used to convert certificates of another format to PEM format.

Converting a certificate in DER encoding to PEM format

A certificate in DER encoding has the extension .crt. To convert for example a vsign3 certificate to PEM format use this OpenSSL command:

- `openssl x509 -inform der -in vsign3.crt -out vsign3.pem`

Converting a client certificate in PKCS12 format to PEM format

A client certificate from PKCS12 format has the extension `.pfx`.

To convert for example a ccapi client certificate to PEM:

1. Type the OpenSSL command:
   ```
   openssl pkcs12 -in ccapi.pfx -out ccapi.pem
   ```
2. If the pkcs12 file is password protected, OpenSSL prompts for the password. Specify the Import Password.
3. When the password is accepted OpenSSL displays the message: `MAC verified OK`
4. OpenSSL then prompts you for a new password (or pass phrase). Specify the new pass phrase.
5. Re-enter the same pass phrase, to verify the pass phrase when OpenSSL prompts you with the message: `Verifying - Enter PEM pass phrase:`
6. When re-entering the password was successful, OpenSSL exits.
7. On failure OpenSSL reports Verify failure.

Viewing the contents of a PEM certificate

You can view the contents of a certificate in PEM format. To view for example the vsign3 certificate, use this OpenSSL command: `openssl x509 -text -noout -in vsign3.pem`

Generating hashed file names of certificates

OpenSSL requires hashed file names of server certificates. The hashed file name consists of a hash obtained from OpenSSL, with a numerical extension starting at 0.

During SSL negotiation OpenSSL receives a server certificate. It calculates the hash of the certificate and uses the hash to find the appropriate certificate. For example, the hash of the vsign3 certificate can be 415660c1. OpenSSL attempts to open the file named 415660c1.0. When successful OpenSSL compares the contents with the received server certificate. If they match OpenSSL continues. Otherwise, OpenSSL attempts to open the file named 415660c1.1, and follows the same procedure etc. If in the end no matching certificate is found, the connection is refused. The hash of a certificate, in this case the vsign3 certificate, can be obtained with this command:

```
openssl x509 -hash -noout -in vsign3.pem
```

Note: Use version 1.0 or later of OpenSSL. The hashes of earlier versions of OpenSSL are incompatible with the version of the OpenSSL library that is linked to the bshell.

The first part 415660c1 of the hashed file name is the hash returned by the openssl command. The extension .0 of the hashed file name is there to support multiple certificates that have the same hash; one of them gets extension .0, the other gets extension .1.

The OpenSSL distribution can contain the c_rehash script. This convenient Perl script can be used to create hashed file names of all certificates in a certain folder. For example, to create hash file names of all certificates in the `$BSE/security/certs/server` folder use this command:

```
c_rehash $BSE/security/certs/server
```

Ensure that the c_rehash script uses OpenSSL version 1.0 or later, otherwise unusable hash names are created. Set the environment variable OPENSSL to use a specific installation of OpenSSL.

Chapter 24: Date and time formatting in LN UI

Date and time formatting depends on several settings.

LN UI can run stand-alone or as a plug-in in Infor Ming.le.

Using LN UI stand-alone

If you use LN UI stand-alone, then the date/time display format is determined by the browser or by the LN user data settings. This depends on the **Custom Date Time Format** setting in the **User Data Template (ttams1110m000)** session:

- If the **Custom Date Time Format** check box is selected, then the date time format that is configured in the user data template is used.
- If this check box is cleared, then the date/time display format is derived from the language preference of your browser. If this is not supported, then the American English date/time display format is used.

Using LN UI within Infor Ming.le

If LN UI runs within Infor Ming.le, then the date/time display format is determined by Infor Ming.le or LN user data settings. This depends on the **Custom Date Time Format** setting in the **User Data Template (ttams1110m000)** session:

- If the **Custom Date Time Format** check box is selected, then the date time format that is configured in the user data template is used.
- If this check box is cleared, then the date/time display format is derived from the **Locale** user setting in Infor Ming.le. If this is not supported, then the American English date/time display format is used.

Date and time formatting in BW

Note: BW is only supported in LN 10.3 and earlier versions.

In BW, formatting of date and time fields is based on the Windows date and time settings and the LN format codes that are linked to the fields. Format codes refer to formats that are defined in the **Date Formats (ttadv4580m000)** and **Time Formats (ttadv4586m000)** sessions. A format determines, for example, whether a date field shows the year with or without the century. Or whether a time field shows only hours and minutes and no seconds.

Date formatting

This section describes how the Windows settings and the LN date formats are combined and used to format date fields in BW.

The properties of the field format are retrieved and verified. If the format is defined to use week numbers and no day number is defined, this format is incompatible with the Windows Date control.

This table shows how the format's properties are changed in that scenario:

Property	Value
Day Number	In Month
Day Name	None
Month Number	Number
Month Name	Abbreviated
Year Number	With Century

Note: If the ERP format specifies that the Day Name or the Month Name must be displayed, the Windows long format is used; otherwise, the Windows short format is used.

This Windows format is then combined with the ERP format. The Windows format determines the order of the individual parts. The ERP format determines how the individual parts are formatted.

This table shows the rules that are applied:

Format d or dd Day number of the month	This format is used if the ERP format uses a day number, otherwise, this format is skipped. This format is mapped to the day number of the month. This is regardless of whether the ERP format uses the day number of the week or the day number of the year. If the **Zero Fill** check box in the **Date Formats (ttadv4180s000)** session is selected, Windows format dd is used, otherwise format d is used.
Format ddd or dddd Day name of the week	This format is used, regardless of whether the ERP format does not use day numbers. If the ERP format uses complete day names, Windows format dddd is used. If the ERP format uses abbreviated names, ddd is used. If the ERP format does not specify which format to use, the current Windows format is used.
Format M or MM Month numbers	This format is used if the ERP format uses a month number; otherwise, this format is skipped. If the **Zero Fill** check box in the **Date Formats (ttadv4180s000)** session is selected, Windows format MM is used; otherwise format M is used.
Format MMM or MMMM Month names	This format is used if the ERP format uses a month name or a month number. If the ERP format uses complete month names, Windows format MMMM is used. If the ERP format uses abbreviated names, MMM is used. If the ERP format does not specify which format to use, the current Windows format is used.

Format yy or yyyy Year numbers	This format is used, regardless of whether the ERP format does not use the year number. If the ERP format uses years with the century, Windows format yyyy is used. If the century is not required, the **Zero Fill** check box determines whether format yy or y is used. Note: When the cursor is in a date field, the year is displayed with the century.
Any literal text between single quotes	The specified text is used.
Any character	This is treated as a date separator. The first time a character such as the period (.) is found in a date, that character is used as the date separator. If different characters are used as date separators, this character is used in place of the other characters. For example, Windows format dd.MM-yyyy becomes dd.MM.yyyy.

Example 1

This table shows the Windows date formats for a Dutch user:

Short Date	d-M-yyyy For example, 9-1-2012
Long Date	dddd d MMMM yyyy For example, maandag 9 januari 2012

A date field uses ERP format 001. This table shows the properties for this format:

Day Name	None
Day Number	In Month
Week Number	None
Month Name	None
Month Number	Number
Year Number	Without century
Zero Fill	Yes

Because no Day Name or Month Name is defined, the date is formatted based on the Windows Short Date format. The date value is formatted as dd-MM-yy, for example, 09-01-12.

Example 2

This table shows the Windows date formats for a user in the United States:

Short Date	M/d/yy For example, 1/9/12
Long Date	dddd, MMMM dd yyyy For example, Monday, January 09, 2012

A date field uses ERP format 002. This table shows the properties for this format:

Day Name	None
Day Number	In Month
Week Number	None
Month Name	None
Month Number	Number
Year Number	With century
Zero Fill	Yes

Because no Day Name or Month Name is defined, the date is formatted based on the Windows Short Date format.

The date value is formatted as: MM/dd/yyyy, for example, 01/09/2012.

Example 3

This table shows the Windows date formats for a user in the United States, the same as in example 2:

Short Date	M/d/yy For example, 1/9/12
Long Date	dddd, MMMM dd yyyy For example, Monday, January 09, 2012

A date field uses ERP format 003. This table shows the properties for this format:

Day Name	None
Day Number	In Month
Week Number	None
Month Name	Abbreviated
Month Number	None
Year Number	With century

Zero Fill	No

Because the abbreviated Month Name must be used, the date is formatted based on the Windows Long Date format. The date value is formatted as: dddd, MMM d yyyy, for example, Monday, Jan 9 2012.

Example 4

This table shows the Windows date formats for a German user:

Short Date	dd.MM.yyyy For example, 09. 01. 2012
Long Date	dddd, d. MMMM yyyy For example, Montag, 9. Januar 2012

A date field uses ERP format 008. This table shows the properties for this format:

Day Name	Complete
Day Number	In month
Week Number	None
Month Name	Complete
Month Number	None
Year Number	With century
Zero Fill	Yes

Because the complete Day Name and Month Name must be used, the date is formatted based on the Windows Long Date format. The date value is formatted as dddd, dd. MMMM yyyy, for example, Montag, 09. Januar 2012.

Time formatting

This section describes how the Windows settings and the LN time formats are used to format time fields in BW.

The data of the field format is retrieved and combined with the Windows time format. The Windows format determines the order of the individual parts; the ERP format determines how the individual parts are formatted.

Note: In all scenarios, the Windows long time format is used.

This table shows the rules that are applied:

Format h or hh 12 hour notation	This is mapped to the Windows hh format.
Format H or HH 24 hour notation	This is mapped to the Windows HH format.
Format m or mm Minutes	If the ERP format uses minutes, this is mapped to the Windows mm format.
Format s or ss Seconds	If the ERP format uses seconds, this is mapped to the Windows ss format.
Format tt AM or PM symbol	This is taken over.
Any character	The first time a character such as the colon (:) is found in a time value, that character is used as the time separator.

Example 1

A user has this Windows long time format: H:mm:ss

A time field uses ERP format 003. This table shows the properties for this format:

Hour Format	12 Hour Format
Minutes	No
Seconds	No

The time value is formatted as HH, for example, 16, which is four o'clock in the afternoon.

Example 2

A user has this Windows long time format: H:mm:ss

A time field uses ERP format 099. This table shows the properties for this format:

Hour Format	12 Hour Format
Minutes	Yes
Seconds	No

The time value is formatted as: HH:mm, for example, 16:24.

Example 3

A user has this Windows long time format: h:mm:ss tt

A Time field uses ERP format 001. This table shows the properties for this format:

Hour Format	24 Hour Format
Minutes	Yes
Seconds	Yes

The time value is formatted as hh:mm:ss tt, for example, 04:24:37 PM.

Date and time formatting in Web UI 10.0 and later

Note: Web UI is only supported in LN 10.3 and earlier versions.

The formatting of date and time fields in Web UI is simpler than in BW. All formatting is done based on the Windows Date and Time settings only. The linked LN format code is not taken into account.

Date formatting uses the Windows Short Date format. These rules apply:

- Format d becomes dd.
- Format M becomes MM.
- Format yy becomes yyyy.

Time formatting uses the Windows Long Time format.

Note: On Linux client PCs, the local date and time conventions are used.

Chapter 25: Document Authorization

Database Change Management (DBCM) supports Document Authorization, using ION Workflow. Document Authorization is about approving or rejecting changes that are made to Business Objects, in a controlled way.

If a user makes changes to a Business Object in ERP Enterprise, these changes must be submitted for approval. Only when these changes are approved, the Business Object may be processed further.

Example

A user changes a Sales Order object, by modifying header data, or by adding, changing, or removing lines. The user must submit the changes for approval. After approval, the Sales Order object can be processed further.

Through DBCM, two versions of the same Business Object can exist during a certain time frame: a checked-in version and a checked-out version.

If a user changes a Business Object, automatically a checked-out version is created. This checked-out version is only visible in maintain sessions for that particular Business Object. This version is unknown to the rest of the ERP system. This checked-out version is a kind of scratch version. A user can change anything. The changes are not available to the rest of the system until the user submits the changes, and someone approves these changes. You can undo these changes, and revert back to the original version.

Administrators can use the **Checked-out Objects (ttocm9599m000)** session to view the objects that are currently checked out, and to perform corrective actions in case of errors. See Actions for checked-out objects on page 193 and the session help.

Modeling and Deploying Document Authorization

To support Document Authorization for a Business Object, a Model must be defined using the Object Change Management modeling sessions (ttocm0101m000). A model defines for which user actions Document Authorization is supported.

A model contains multiple object types. Per object type, this information is specified:

- Actions
- Table relations, if the object contains data from multiple tables
- Mappings

Infor delivers a predefined model. You can copy this model and customize it to your needs. You can choose for which Business Objects you want to use Document Authorization, by specifying this in a Deployment.

If the Model describes two Object Types A and B, and you use Document Authorization for Object Type A. DBCM does not create checked-out versions for instances of Object Type B. Instances of Object Type A are always checked-out.

Depending on the active user actions in the Deployment, one of these events occurs when changes are submitted:

- To get approval, the LN application logic publishes a Workflow BOD to ION.
- The object is automatically checked in.

Prerequisites

To support Document Authorization, these prerequisites must be met:

- ION Workflow must be installed.
- The workflow definitions that are specified in the actions of the object types must exist in ION Workflow.
- The LN application must use the DAL2 concept. The DAL of the root table must implement these hooks:
  - on.submit()

    This hook is executed when changes that are made are submitted for approval.
  - on.recall()

    This hook is executed when the submit must be recalled. Both these hooks must publish a Workflow BOD to ION Workflow.
- The LN application must be adapted.

Application changes

To support Document Authorization, the application must be adapted.

After a Business Object is checked-out, no changes to any related Business Objects of another Object Type are allowed. Therefore the DAL may only update tables that belong to the Business Object itself. Only during checking-in, Business Objects of other Object Types may be updated. To support this, you must use the `dbcm.object.is.being.checked.in()` function.

If a session can be used to change data of a Business Object for which Document Authorization must be supported. This session must specify the selected Object Type. In that way the portingset knows whether to include checked-out versions when selecting data from the database. For maintain sessions this is done automatically by the 4GL Engine, based on the main table. For update sessions, you must use these funtions to achieve this: `dbcm.select.object.type()` or `dbcm.select.object.instance`.

The session must indicate the user action that is being performed. For example, if the user presses the **Release to Warehousing** button, and this supports Document Authorization. The application must indicate that this action is being performed by the application logic. DBCM can then determine whether this action requires approval.

The 4GL Engine supports these standard actions through the User Interface:

- Inserting

- Updating
- Deleting

Any other, application-specific, actions must be selected by the application using the `dbcm.select.object.action` function.

Checked-out Business Object states

This table shows the states a Business Object can have:

State	Description
Draft	The object is checked-out, it can be modified and any changes can be submitted. Any changes can be undone by performing a Revert to Approved in the User Interface.
Draft (Revision)	The object is in the Draft state, for a second time; this state is equal to the Draft state, except that an object can only enter this state after a Recall of any submitted changes was successful.
Pending	Any changes to the object have been submitted and the user must wait until the changes are Approved or Rejected. The object cannot be modified.
Recall Requested	The user made a request to ignore any submitted changes, because the user, for example, wants to make more changes to the object. The object cannot be modified.
Rejected	Any submitted changes to the object were not approved. The user must either make other changes and re-submit them, or perform a Revert to Approved. The object can be modified.
Approval Received	Usually this state is not visible to the user. It can only be visible if somehow, after receiving an Approval, the object cannot be checked-in. In this situation an administrator must force a check-in, or discard any changes and perform a Revert to Approved. The object can be modified.
Approved	Any submitted changes to the object have been Approved, and the object has been checked-in. The object can be modified.

Importing the standard model

1 Start the **Models (ttocm0101m000)** session.

2 On the *appropriate* menu, select **Import Standard Model**.

Creating a model

To create your own model, perform these tasks:

1. Duplicate the standard model.
2. Add object types to the new model.
3. Add actions to the object types.
4. If the object consists of multiple tables, add table relations to the object type.
5. Add mappings to the object type.
6. Validate the model.

See these sections.

Upon completion of this procedure, you must deploy the model. See Deploying a model on page 192 .

Note: When you customize a model, you probably must modify scripts and libraries. See the *Infor ES Programmer's Guide*.

Duplicating the standard model

To duplicate the standard model:

1. Start the **Models (ttocm0101m000)** session.
2. Select the standard model and click **Duplicate**.
3. Specify the model code and description for the new model.
4. Save the new model.

Adding object types to the new model

You can add multiple object types to the model.

To add an object type to the new model:

1. In the **Models (ttocm0101m000)** session, select the new model.
2. On the *appropriate* menu, select **Object Types**. The **Object Types (ttocm0102m000)** session is started.
3. Add an object type. See the session help.
4. Save the new object type.

Adding actions to the object types

You can add one or more actions to each object type.

To add an action an object type:

1. In the **Object Types (ttocm0102m000)** session, navigate to the object type.
2. On the **Actions** tab, click **New Action**.
3. Add an action. See the session help.
4. Save the new action.

Adding table relations to the object types

If the object contains data from multiple tables, you must specify table relations between the tables that are used in the object.

For example, the sales order object is based on the Sales Orders (tdsls400) and Sales Order Lines (tdsls401) tables. An order header without lines is useless. Therefore, for the sales order object type, specify a relation from the Sales Order (orno) field in the tdsls401 table to the tdsls400 table.

To add a table relation to an object type:

1. In the **Object Types (ttocm0102m000)** session, navigate to the object type.
2. On the **Table Relations** tab, click **New Table Relation**.
3. Add a table relation. See the session help.
4. Save the new table relation.

Adding mappings to the object types

You can add one or more mappings to each object type.

To add a mapping to an object type:

1. In the **Object Types (ttocm0102m000)** session, navigate to the object type.
2. On the **Mappings** tab, click **New Mapping**.
3. Add a mapping. See the session help.
4. Save the new mapping.

Validating the model

To validate the model:

1. In the **Models (ttocm0101m000)** session, select the new model.
2. On the *appropriate* menu, select **Validate**. A message, indicating whether the validation was successful, is displayed.
3. If errors occurred, view the log file and solve the problem.

Deploying a model

1. Generate a deployment.
   a. In the **Models (ttocm0101m000)** session, select the model.
   b. On the *appropriate* menu, select **Deploy...**. The **Deploy Model (ttocm0101m100)** session is started.
   c. Specify a description for the deployment and specify the package combination to which the deployment must be linked.
   d. Click **Deploy**. The deployment is generated.
2. Deploy the actions of the deployment.
   a. Start the **Deployments by Package Combination (ttocm0111m000)** session.

b Select the deployment and, on the *appropriate* menu, select **Deployed Actions**. The **Deployed Actions (ttocm0112m000)** session is started.

c Enable the desired actions. To enable an action, select the **Enabled** check box in the corresponding row. To enable all actions directly, on the *appropriate* menu, select **Enable All**.

d Save the changes and close the **Deployed Actions (ttocm0112m000)** session.

3 Activate the deployment.

a Ensure all users are logged off.

b In the **Deployments by Package Combination (ttocm0111m000)** session, select the deployment.

c On the *appropriate* menu, select **Activate**.

Conversion/reconfiguration indicators are updated and the **Convert to Runtime Data Dictionary (ttadv5215m000)** session is started. This session automatically updates the table definitions and reconfigures the related tables for the concerning package combination(s). The **Convert to Runtime Data Dictionary (ttadv5215m000)** session is started without UI.

d Log off and log on again.

The changes are now fully actualized. All users can log on again.

Actions for checked-out objects

In the **Checked-out Objects (ttocm9599m000)** session, you can perform these actions on checked-out business objects:

- **Check In**
- **Recall**
- **Revert To Approved**
- **Submit**

See the online help of the **Checked-out Objects (ttocm9599m000)** session.

Chapter 26: Data Upgrade Engine

The *Data Upgrade Engine* (DUE) is used to update the customer data after a *Feature Pack* (FP) upgrade.

To perform a data upgrade, you must execute a *Data Upgrade Run*. In some situations, LN automatically generates a data upgrade run. You can also manually create data upgrade runs.

Each data upgrade run belongs to a particular package combination and performs a data upgrade for one or more companies that are linked to that package combination.

A data upgrade run consists of *upgrade tasks*. Each task executes one *upgrade program* for one company. Each upgrade program upgrades a particular table.

Example

Companies 410 and 411 are linked to package combination b61ua6. Data upgrade run 19 is linked to package combination b61ua6 and contains these upgrade tasks:

Company	Upgrade program
410	tcsptdll0901 - Upgrade of table tcfin020
410	tcsptdll0902 - Upgrade of table tcfin015
410	tcsptdll0905 - Upgrade of table tccom000
410	tcsptdll1902 - Upgrade of table tccom710
411	tcsptdll0901 - Upgrade of table tcfin020
411	tcsptdll0902 - Upgrade of table tcfin015
411	tcsptdll0905 - Upgrade of table tccom000
411	tcsptdll1902 - Upgrade of table tccom710

Note: You do not have to upgrade all companies of a package combination in one run. You can define multiple runs for different companies of the same package combination. For example, you can define different runs for live companies and for test companies, and execute these runs at different times. See the next example.

Example

These companies are linked to package combination A:

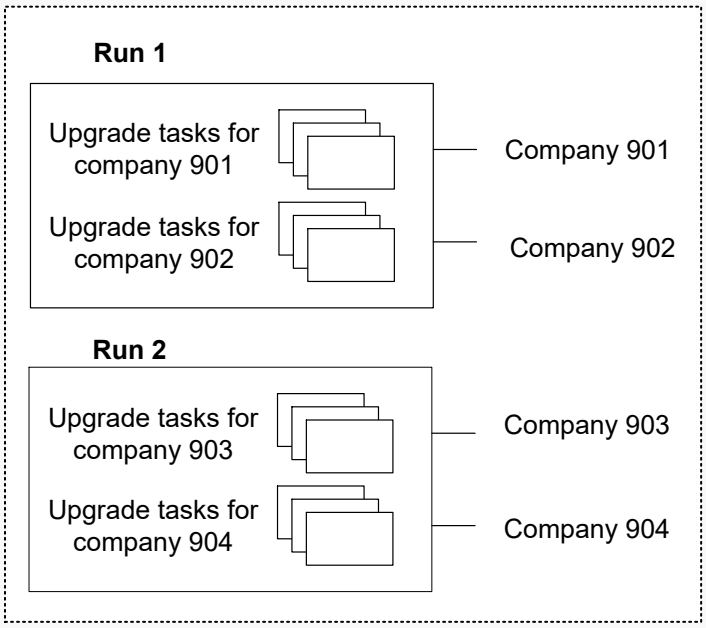
- 901 - live data
- 902 - live data
- 903 - test data
- 904 - test data

You define these runs for package combination A:

- 1 - to upgrade data for company 901 and 902
- 2 - to upgrade data for company 903 and 904

See this figure:

Data Upgrade Runs defined for Package Combination A*



* Package combination A is linked to companies 901, 902, 903, 904

Data upgrade runs are executed in these situations:

Situation	Description	Note
1	You installed a new feature pack (*service pack*) in a package combination.	A new data upgrade run is generated automatically after this action. You must initialize and execute this data upgrade run.

Situation	Description	Note
2	You moved companies to another package combination through the **Change Package Combination by Company (ttaad1101m000)** session.	A new data upgrade run is generated automatically after this action. You must initialize and execute this data upgrade run.
3	In a package combination, you replaced a VRC by another VRC. For example, in package combination A, you replaced VRC tdXXXX by tdZZZZ.	An upgrade is only required if this change affects tables. In that case, you must manually create and execute a data upgrade run. Tables are affected, for example: • If the new VRC belongs to a new feature pack, and therefore contains changed table definitions • If the new VRC is derived from a VRC that belongs to a new feature pack
4	You changed the VRC derivation structure.	An upgrade is only required if this change affects tables. In that case, you must manually create and execute a data upgrade run. Tables are affected, for example, if you derive a VRC from a VRC that belongs to a new feature pack. For example: • Package combination A contains VRC tdYYYY, which is derived from an FP5 VRC. You change the derivation, so that tdYYYY is derived from an FP6 VRC, which contains changed table definitions. • You must create and execute a data upgrade run for package combination A.

Note:

If a data upgrade run is generated, or created manually, the involved companies are locked. Normal users cannot access these companies anymore. In the **Companies (ttaad1100m000)** session, these companies have the **Upgrade Needed** status.

You can start a run in simulation mode: the Data Upgrade Engine will run the upgrade without actualizing the transactions.

Execution

During a data upgrade run, various upgrade programs are executed. Some upgrade programs must have finished successfully before another program can start.

To enhance the performance of the data upgrade process, you can run the DUE using multiple bshells. In this case multiple CPUs can be utilized. The DUE gathers the list of upgrade programs that must be executed and spreads the work over the different bshells. When you start a data upgrade run, you can specify the number of bshells you want to activate.

The sequence in which the upgrade programs are executed depends on:

1. Upgrade Batch

   Each upgrade program belongs to one of these batches:

   - **First**
   - **Middle**
   - **Last**
   - **Unlocked**

   During a data upgrade run, first the upgrade programs of the **First** Batch are executed. Then the **Middle** Batch is executed. The **Last** Batch is executed after the **Middle** Batch. When the **Last** Batch is executed, the companies are released and the **Unlocked** Batch is started. When the companies are released, users can start using the companies again.

2. Prerequisite dependencies within batches

   An upgrade program can depend on other upgrade programs in the same batch, which must have been executed successfully before the current program can start.

   An upgrade program can only be started if its pre-requisites are executed.

3. Runtime Class

   If a batch contains multiple upgrade programs whose pre-requisites are executed, these programs are started based on their runtime class.

   Each upgrade program belongs to one of these runtime classes:

   - **Huge**
   - **Large**
   - **Medium**
   - **Small**
   - **None**

   First the upgrade programs with runtime class **Huge** are executed. Then the **Large** programs are executed. Subsequently the **Medium** programs are executed. Finally the **Small** programs are executed.

   Upgrade programs with Runtime Class **None** do not have to be executed, but can be a prerequisite for other programs.

   A task can have a higher runtime class than its prerequisite. During execution, the prerequisite gets this higher class.

To create and execute a data upgrade run, see Executing a data upgrade run on page 198

Simulation mode

You can start a run in simulation mode. During a simulation, the Data Upgrade Engine runs the upgrade without actualizing the transactions. This is useful for testing purposes.

See the online help of the **Data Upgrade Engine (ttspt2201m000)** session.

Performance and finetuning

You can optimize the performance of the Data Upgrade Engine. See Performance and finetuning.

Troubleshooting

If one or more upgrade programs fail during a data upgrade run, use the troubleshooting procedure. See Troubleshooting.

Error Recovery

Each upgrade program runs in a separate process. The upgrade programs can return status information about the success or failure of the execution. Some statuses interrupt a part of the upgrade process. Other statuses only inform you that something went wrong, or that the upgrade program was executed successfully. If a fatal error occurs, such as error 606, "reference not found", only upgrade programs that are dependent on the failing upgrade program cannot be executed anymore. All other upgrade programs can still be executed.

Logging

Upgrade programs can generate reports of errors and changes. When you start a data upgrade run, you can indicate that only errors should be reported.

You can access log information from the **Data Upgrade Tasks (ttspt2520m100)** session.

Executing a data upgrade run

How to create and execute a *Data Upgrade Run*.

1 Display the run information

- **a** Start the **Data Upgrade Runs (ttspt2500m000)** session.
- **b** Go to the **Run Information** tab.

2 If the data upgrade run already exists, initialize the run

Perform this step only if the data upgrade run you want to execute already exists, for example because it was generated automatically.

To initialize an existing run:

- **a** Select the run and on the *appropriate* menu, select **Initialize Data Upgrade Run**. The **Initialize Data Upgrade Run (ttspt2200m000)** session starts.
- **b** Fill out the fields in the session. Among other things, specify this information:
  - If applicable, an **Upgrade Task Source**

    For example, you can specify a *PMC Solution* on which the initialization must be based.
  - The companies you want to upgrade

  The **Source Feature Pack** is already filled in.

  Caution: Ensure the correct source feature pack number is specified. The data upgrades to be performed is based on the specified source feature pack number. If the number is too low, too much data is converted. Possibly data is overwritten with incorrect values. If the number is too high, not all data is converted. Possibly data is not consistent.

Note:

- The initialization process assigns the specified feature pack number to all *upgrade tasks* in the run.
- After an incorrect initialization, you can correct the **Source Feature Pack** for the upgrade tasks/ *upgrade programs* in the run. To perform this correction, use the **Global Change Source Feature Pack (ttspt2220m000)** session.

c Click **Initialize**.

d When the initialization is finished, a "Process completed" message is displayed. To remove the message box, click **OK**.

e Close the **Initialize Data Upgrade Run (ttspt2200m000)** session.

3 Perform this step only if the data upgrade run does not exist yet.

To create and initialize a new run:

a In the **Data Upgrade Runs (ttspt2500m000)** session, click **New**. The **Initialize Data Upgrade Run (ttspt2200m000)** session starts.

b Fill out the fields in the session. Among other things, specify this information:

- If applicable, an **Upgrade Task Source**

  For example, you can specify a *PMC Solution* on which the initialization is based.

- The companies you want to upgrade
- The number of the **Source Feature Pack**, such as 6 in case of FP6

Caution: Ensure the correct source feature pack number is specified. The data upgrades to be performed is based on the specified source feature pack number. If the number is too low, too much data is converted. Possibly data is overwritten with incorrect values. If the number is too high, not all data is converted. Possibly data is not consistent.

Note:

- The initialization process assigns the specified feature pack number to all *upgrade tasks* in the run.
- After an incorrect initialization, you can correct the **Source Feature Pack** for the upgrade tasks/ *upgrade programs* in the run. To perform this correction, use the **Global Change Source Feature Pack (ttspt2220m000)** session.

Note: if you select the **Start Data Upgrade Engine After Initialization** check box, the run starts automatically after the initialization is finished. You cannot check the source feature pack or finetune the run before it is started.

c Click **Initialize**.

d When the initialization is finished, a "Process completed" message is displayed. To remove the message box, click **OK**.

e Close the **Initialize Data Upgrade Run (ttspt2200m000)** session. The new run is displayed in the **Data Upgrade Runs (ttspt2500m000)** session.

4 Finetune the run.

To finetune the run, you can select these actions on the *appropriate* menu in the **Data Upgrade Runs (ttspt2500m000)** session:

Companies by Data Upgrade Run	Starts the **Companies by Data Upgrade Run (ttspt2510m000)** session. By default, all companies of a package combination are processed in the same run. To split the upgrade process, you can move companies to other runs. In this way you can, for example, process a package combination's live companies and archive companies in two separate runs. To move companies to another run: **a** In the **Companies by Data Upgrade Run (ttspt2510m000)** session, select the companies you want to move. **b** On the *appropriate* menu, select **Move Companies to Another Run**. The **Move Companies to Another Run (ttspt2210m000)** session starts. **c** Enter the required information: • Optionally, change the company selection. • Specify the destination run. You can create a new run, or select an existing run. **d** Click **Move**. Note: companies that are related to each other must be linked to the same run. You can only move a company to another run, if you also move its related companies. If the company selection does not contain all related companies, an error message is displayed. The message specifies the missing companies that must be included in the selection. Click **Make Valid** to automatically select the missing companies.
Data Upgrade Tasks	Starts the **Data Upgrade Tasks (ttspt2520m000)** session. Double-click an *upgrade task* to start the **Data Upgrade Tasks (ttspt2520m100)** details session. Use this session to change, for example, these properties of an upgrade task: • The **Source Feature Pack** • The **Runtime Class** **Note** • Set the **Runtime Class** to **`Huge`**, **`Large`**, **`Medium`**, or **`Small`**. • The **Real Run Status** **Caution:** Only change this status in case of troubleshooting. See Troubleshooting. If you set the **Real Run Status** to **`Released`**, the task is not processed. In this way you can skip tasks that would fail during execution, and ensure the run succeeds anyway. This can cause unpredictable results or even data corruption. Therefore setting the **Real Run Status** to **`Released`** is at your own risk. For more information, refer to the session help.

5 Execute the run

a Select the run in the **Data Upgrade Runs (ttspt2500m000)** session.

b On the *appropriate* menu, select **Data Upgrade Engine**. The **Data Upgrade Engine (ttspt2201m000)** session starts.

c Fill out the fields in the session and click **Continue**. See the session help.
Note: If a task fails during the execution of the run, an error message is displayed in the progress bar. In case of a fatal error, the failing upgrade program stops. All dependent upgrade

programs cannot be executed anymore. The run continues to execute all other upgrade programs.

You can solve the problem that caused the error and start the failed task again. See Troubleshooting on page 203.

Performance and finetuning

This section describes how you can optimize the performance of the *Data Upgrade Engine* (DUE).

The performance of the DUE is influenced by:

- The runtime class of the upgrade tasks. See Runtime class on page 201.
- The usage of additional servers. See Additional servers on page 202.
- The usage of the local server for processing. See Using the local server for processing on page 202.
- The usage of sub-tasks. See Using sub-tasks on page 203.

You can use the Call Graph Profiler to identify potential performance bottlenecks. See Using the Call Graph Profiler on page 203.

Runtime class

During initialization of the DUE run, the upgrade programs determine their runtime class. This is based on factors such as parameter settings, table sizes, and source feature pack. Common rules are used. For your specific environment there can be factors that lead to a not optimal determination of the runtime class. Long running tasks can be scheduled later than small tasks. This causes a longer total elapsed time.

We recommend to perform a test upgrade run before you start the real run. You can analyze the results of the test run, and optimize the running class where necessary. Complete these steps:

1. Perform a test run. See Executing a data upgrade run on page 198.

   For a realistic test, perform this test run on companies that are copied from the real live data companies.
2. Print and analyze the results of the test run.
   a. Start the **Print Data Upgrade Run Information (ttspt2400m000)** session.
   b. Select the **Data Upgrade Run Information [Flat File]** report and print the data of the test run to a file.
   c. Optionally, import this file in MS Excel.
   d. Analyze the results. Per upgrade task, view the runtime class and the duration.
3. Optimize the running class in these situations:
   - A task, which lasts long, has a low running class.

     Select a higher running class for the task.
   - A task, which takes a short time, has a high running class.

Select a lower running class for the task.

For example:

- A task has running class **Medium** and lasts 3 hours. Change the running class to **Large** or **Huge**.
- A task has running class **Huge** and lasts only 30 minutes. Change the running class to **Large** or **Medium**. Other tasks can take advantage of this.

The duration of an upgrade task differs per company because it depends on many variables. Therefore, for each company or set of companies, investigate the maximum duration of an upgrade task before you change the runtime class.

Additional servers

Caution:

If the whole system capacity can be used for the DUE, we recommend to specify one additional bshell per CPU. For example, if your LN server contains 8 CPUs, specify 8 additional bshells.

The DUE is high CPU intensive and other users are hampered when the number of additional upgrade tasks is too high. Therefore reduce the number of additional bshells if not the whole system can be used for the DUE.

In the **Number of Additional Servers** field in the **Data Upgrade Engine (ttspt2201m000)** session, you can specify the number of additional bshells that are used to run the Data Upgrade Engine.

Additional bshells can improve the performance of the DUE because several upgrade tasks can be started in parallel.

The additional bshells perform many I/O actions. It is not recommended to add more bshells than the I/O subsystem can handle. If you do not know the I/O capacity of the I/O sub system, as a rule of thumb, specify one additional bshell per CPU.

Note: If you set the **Number of Additional Servers** field to 0, the DUE is run in a single bshell.

Using the local server for processing

When you use additional bshells, the local bshell , where the **Data Upgrade Engine (ttspt2201m000)** session is started, schedules the upgrade tasks. This local bshell gathers the list of upgrade programs that must be executed and spreads the work over the different bshells.

If you select the **Use the local Server for processing** check box in the **Data Upgrade Engine (ttspt2201m000)** session, the local bshell schedules the upgrade tasks and runs the upgrade tasks.

We recommend these options:

- Select the **Use the local Server for processing** check box if you select 1 additional bshell in the **Number of Additional Servers** field.
- Clear the **Use the local Server for processing** check box if you select more than 1 additional bshell in the **Number of Additional Servers** field. The local bshell only schedules the upgrade tasks. If the local bshell also had to run upgrade tasks, it might be too busy to schedule tasks for the different additional bshells in time.

Using sub-tasks

Some long running *upgrade tasks* have sub-tasks. These sub-tasks are defined in the upgrade programs delivered by Infor and are created during the initialization of a DUE run. Each sub-task has its own data range to perform a part of the upgrade task. To improve the performance of the DUE, sub-tasks can be run in parallel in different bshells. If a task is sub-task enabled, you can manually add, change, or remove sub-tasks. See the online help of the **Sub-Tasks (ttspt2535m000)** session. This session runs in a tab in the **Data Upgrade Tasks (ttspt2520m100)** details session.

Using the Call Graph Profiler

To identify potential performance bottlenecks, you can run the Data Upgrade Engine with the Call Graph Profiler activated.

To view the generated files, use the **Files (ttspt2530m000)** session. The names of the generated files have this format:

```
profile.[bshell.pid].ttspt2203m000.[pid].html
```

To use the Call Graph Profiler, start the **Debug and Profile 4GL (ttadv1123m000)** session and select the **Enable Call Graph Profiler** check box.

For details on the Call Graph Profiler, see these documents:

- *Infor LN - Performance, Tracing and Tuning Guide*
- *Infor ES Programmer's Guide*

Troubleshooting

Use this procedure if one or more *upgrade programs* fail during a *Data Upgrade Run*.

1. Start the **Data Upgrade Runs (ttspt2500m000)** session.
2. Select the run and, on the *appropriate* menu, select **Data Upgrade Tasks**. The **Data Upgrade Tasks (ttspt2520m000)** session starts.
3. Select **View > Sort By > Upgrade Tasks by Real Status**.
4. Navigate to the *upgrade tasks* with **Real Run Status `Failed`**.
5. Double-click a task. The **Data Upgrade Tasks (ttspt2520m100)** details session starts.
6. In the **Files by Data Upgrade Task** tab, double-click a log file.
7. View the contents of the log file.
8. Solve the problem that caused the failure. For example:
   - Create a correction program.
   - Contact Infor Support.
   - Contact a consultant.
9. If the problem is solved, ensure the task is restarted.
   - If the Data Upgrade Run is still running, set the **Real Run Status** of the task to **`Ready for Retry`**. You must perform this action in the **Data Upgrade Tasks (ttspt2520m100)** details session.

- If the Data Upgrade Run does not run anymore, restart the run. Ensure the failed task is started:
  a Select the run in the **Data Upgrade Runs (ttspt2500m000)** session.
  b On the *appropriate* menu, select **Data Upgrade Engine**. The **Data Upgrade Engine (ttspt2201m000)** session starts.
  c Select the **Include failed upgrade tasks** check box.
  d Fill out the remaining fields and click **Continue**.

Trace level

You can set an additional **Trace Level** in the **Data Upgrade Engine (ttspt2201m000)** session.

The trace level indicates the amount of information that is written to this file:

`$BSE/log/log.due`

The information provides insight in the scheduling and the communication between the multiple bshells.

You can select these trace levels:

- **0 - No Extra Logging**
- **1 - Some Details**
- **2 - More Details**
- **3 - Full Details**

Performance and finetuning

You can optimize the performance of the Data Upgrade Engine, see "Performance and finetuning".

Chapter 27: Application Personalization

You can personalize the application to your own preferences.

Ensure to select the **Allow Application Personalization** check box in your user data template properties on the LN server (User Data Template (ttams1110m000) session).

Use these sessions to personalize your application:

- Session Personalizations (ttadv9100m000)

  In this session, the personalizations made in Web UI are displayed as personalizations on 'user' level for the specific user.
- Conditional Formatting (ttadv9502m000)

  You can define conditions to apply special formatting effects to the data displayed in LN sessions.
- Menu Personalizations (ttadv9505m000)

  To maintain the menu personalizations for LN UI users.
- Export Personalizations (ttaad7201m000)

  To export miscellaneous types of configuration data.
- Import Personalizations (ttaad7202m000)

  To import miscellaneous types of configuration data as exported by the **Export Personalizations (ttaad7201m000)** session.

For details about personalizations, use the online help in the sessions or see *Infor Ming.le-LN Plug-in - User Guide (LN UI)*

Role-based personalizations

You can define and apply personalizations on role level.

These levels exist:

- User
- Role
- Enterprise Modeler roles
- Company

The created personalizations at role level, not to be confused with Enterprise Modeler roles, are designed to function in a non-Enterprise Modeler context. You can create Roles and assign them to users. Role-based personalization is enabled for a user if a default role is specified for this user. A role-enabled user has one current role and can have multiple roles assigned. A user can change his current role to any other role that is assigned to the user.

To support this functionality use these sessions:

- **Roles (ttaad1150m000)**
- **User Roles (ttaad2104m100)**
- **Change Role (ttdsk2009m000)**

Menu personalization

LN UI users can personalize the LN navigation menus that are displayed in the application navigation menubar and the LN Navigator.

LN UI users can perform these menu personalizations:

- Hide menu items.
- Unhide menu items.
- Change the order of the items in a menu.

Select the **Allow Application Personalization** check box in the user data template to be able to personalize menus. Users can maintain their menu personalizations through the **Personalize Menu (ttadv9205m000)** session.

To maintain the menu personalizations of all users, use the **Menu Personalizations (ttadv9505m000)** session. In this session you can perform these actions:

- Edit menu personalizations
- Remove personalizations
- Copy menu personalizations to a user, a role or a company.

For details about personalizations, use the online help in the sessions or see the *Infor LN UI User Guide*.

Graph personalization

You can personalize, define and maintain graphs for sessions with the **Personalize Graphs on Form (ttadv9130m100)** session.

The session consists of two parts: Graphs and Arguments. In the Graphs section you define the graphs for a session. You can define multiple graphs per session.

Select “Export/Import Graphs” from the **Actions** menu to import the standard graphs. The definition of the standard graphs are required to show the graphs in the sessions.

Note: The Arguments section is not applicable for tenant systems.

In the Arguments section you can define the arguments for a graph. Ensure that you have selected a graph in the Graphs section before specifying Arguments.

To set a graph entry as default:

1. Select the graph entry to be made as the default
2. Click "Set as Default".

To Update Project or Report Catalog for one or more graph entries.

Without selected graph entries:

1. Select “Update or Project Catalog” from the **Action** menu.
2. Specify the range of sessions on which to perform the update.
3. Specify the range of graphs on which to perform the update.
4. Specify the range of sequence numbers on which to perform the update.
5. Specify the new value for Project. If blank, Project is not updated for the graphs within the selection.
6. Specify the new value for Report Catalog. If blank Report Catalog is not updated for graphs within the selection.
7. Click **Update** to perform the update process.

With selected graph entries:

1. Specify the new value for Project. If blank, Project is not updated for the graphs within the selection.
2. Specify the new value for Report Catalog. If blank Report Catalog is not updated for graphs within the selection.
3. Click **Update** to perform the update process.

Chapter 28: Digital signatures

A digital signature is an electronic signature that can be used for these purposes:

- To authenticate the identity of the sender of a message or the signer of a document.
- To ensure that the original content of the message or document that has been sent is unchanged.

Digital signatures are easily transportable, cannot be imitated by someone else, and can be automatically time-stamped. A digital signature can be used with any kind of message, whether it is encrypted or plain text.

Digital signatures provide these features:

- Authentication

  Digital signatures are used to authenticate the source of messages. The ownership of a digital signature key is bound to a specific user. Therefore, a valid signature shows that the message was sent by that user.

- Integrity

  In many scenarios, the sender and receiver of a message require assurance that the message has not been changed during transmission. To provide this feature, digital signatures use cryptographic message digest functions.

- Non-repudiation

  Digital signatures ensure that senders who have signed the information cannot, at a later stage, deny they have signed it.

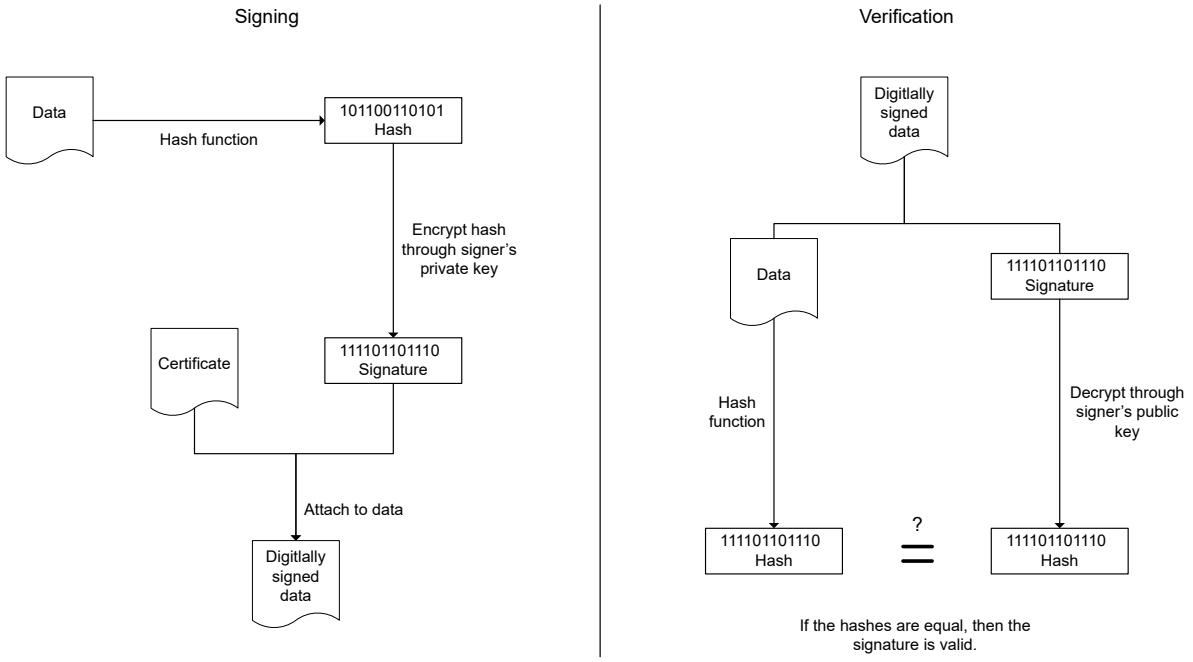
Simplified overview of how digital signatures work

Digital signatures require an asymmetric key pair that consists of a public key and a private key.

These keys are large numbers that are mathematically related. These keys are used to encrypt and decrypt documents. The private key is kept confidential with the owner. It is usually stored on secure media, such as a crypto smart card or a crypto token. The public key is shared with everyone.

To digitally sign an electronic document, senders use their private key (encryption). To verify the digital signature, the recipients use the sender’s public key (decryption).

This diagram shows how digital signatures ensure authenticity, integrity and non- repudiability:



Example

This simplified example shows how the concept of digital signatures works.

You want to send a contract to a business partner. You want to give your business partner this assurance:

- The contract has not been changed during transmission.
- The contract originates from you.
- The contract is sent at a certain moment in time.

To create the contract so that it holds a digital signature:

1. Create a contract in LN.
2. Print the contract to get the electronic form of the contract as a document, for example as a PDF file.
3. Use special software to create a message digest, also known as 'hash', of the contract.
4. Use your private key to encrypt the message digest.
5. The encrypted message digest becomes your digital signature of the contract and is appended to the contract
6. Optionally, also your Digital Signature Certificate (DSC) is appended to the contract for your business partner to speed up the validation of the contract.

At the other end, your business partner receives the contract, for example through an email message:

1. To validate that the contract is intact and originates from you, your business partner generates a message digest of the received contract.
2. Your business partner uses your public key to decrypt the digital signature that was received with the contract. The public key is obtained from the embedded DSC or through the public channels.

3 If the message digest that is generated from the digital signature matches the message digest that was generated in step 1, the integrity of the received contract is verified.

Message digest

A message digest is also known as the hash of a message. This is a small piece of data that is generated when a specific mathematical calculation, a hashing function, is applied on that message.

Message digests have these important characteristics:

- A small alteration in the original message causes a big change in the message digest.
- You cannot derive the original message from the message digest. The message digest that is produced from these functions is a fixed length bit string. The widely used SHA -1 message digest function: generates a 160-bit message digest. The SHA-2 function generates a 256-bit message digest as output.

Public key infrastructure

PKI is the acronym for Public Key Infrastructure.

The technology works with a pair of keys, one of which is made public and the other is kept secret. Therefore, the technology is called Public Key Cryptography. The secret key is usually called the private key. Everyone has access to the public key. Therefore, users can start secure communications without previously sharing a secret through another medium with their correspondent.

PKI is the underlying system that is required to perform these actions:

- Issue key pairs and certificates to applicants.
- Publish the public information through public keys, which are used to establish trust.

PKI is a combination of software, encryption technologies, and services. Through these services, enterprises can protect the security of their communications and business transactions over networks. This is achieved by attaching digital signatures and digital signature certificates.

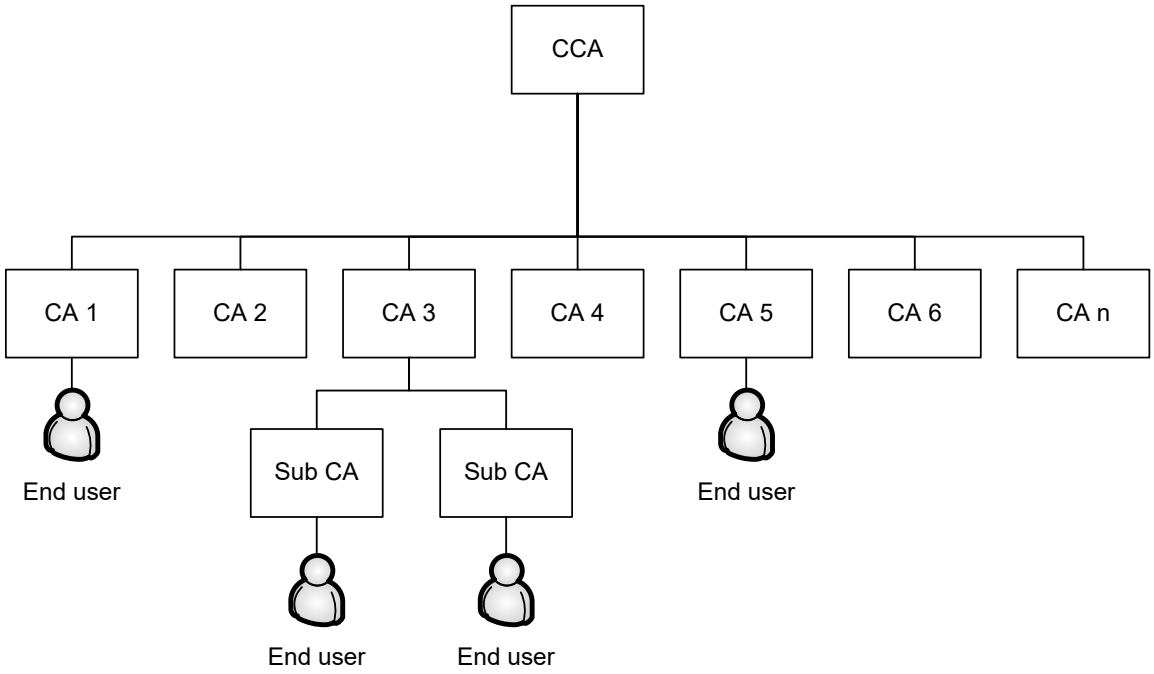
This system relies on mutually trusted third parties to perform these actions:

- Verify the identity of a person or organization.
- Attach that identity to a particular public key.

Using a third party for trust allows end users to delegate trust, instead of having to verify each certificate that is used by another party.

A PKI usually consists of one Central Certifying Authority (CCA) per country. This CCA is usually controlled by the government, and linked to one or more Certifying Authorities (CA’s). The CA's in turn can contain a Registration Authority (RA) and a Validation Authority (VA).

This diagram shows a sample PKI hierarchy:



Digital signature certificates

Certificates serve as identity of an individual for a certain purpose. For example, a driver's license identifies someone who can legally drive in a given country.

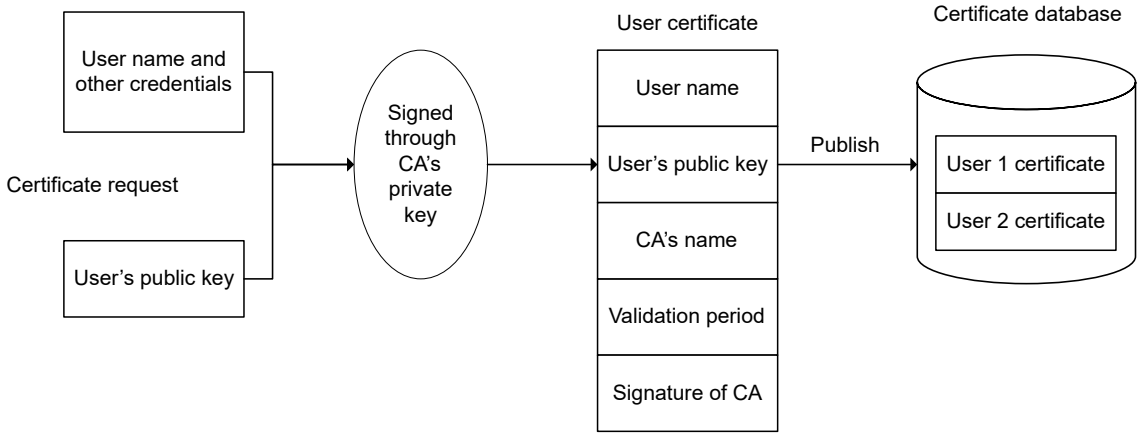
Likewise, a Digital Signature Certificate (DSC) can be presented electronically to prove your identity or your authority to access information or services on the internet.

A DSC is an electronic document that uses a digital signature to bind together a public key with an identity. An identity contains information, such as the name of a person or an organization, or their address. The certificate can be used to verify that a public key belongs to the individual. Digital certificates are the digital equivalent, that is, electronic format, of physical or paper certificates. Examples of physical certificates are driver's licenses, passports, or membership cards.

Digital Signature Certificates are endorsed by a trusted authority that is empowered by law to issue them. This authority is known as a Certifying Authority (CA). The CA is responsible for vetting all applications for Digital Signature Certificates. When satisfied, the CA generates a digital certificate. To achieve this, the CA uses its own private key to digitally sign the public key of the individual along with other information.

To verify authenticity, the recipient of a digitally signed document determines whether the digital signature certificate is valid, and whether the issuer is trusted. If the issuer is trusted, the identity that is attached to the certificate can be trusted. Therefore, the signature belongs to the person or organization that signed the document.

This diagram shows an overview of digital signature certificates:



To configure the storage of keys and the trusted certificates, use the **Digital Signing Configuration (ttaad2655m000)** session.

For information on how to sign documents that are generated in LN, see Implementing digital signatures with Document Output Management on page 213.

Trusted lists

Trusted lists provide a mechanism to establish trust for legal purposes.

The CCA of a country can publish a trusted list at a known URL; this list contains all CA's that are trusted by the CCA of that country. The list is trusted because the list itself is also signed by the CCA, using its own private (trusted) key.

Timestamps

To prevent repudiation and discussion about the date a document was signed, you can add a trusted timestamp.

The timestamp is provided by an independent, trusted third party; often a CA that serves a double purpose. Because the CA is trusted, also the provided timestamp is trusted.

Implementing digital signatures with Document Output Management

To implement digital signatures to sign or seal documents using Document Output Management:

1. Specify the configuration settings for digital signing
   a. In the **Digital Signing Configuration (ttaad2655m000)** session, specify the parameters that define where keys can be found. If you do not use trusted lists, also specify the path for the trusted certificate store.
   b. If you use trusted lists, add these in the **Trusted Lists (ttaad2557m000)** session. Then, to refresh the trust store, run the **Refresh Trusted Lists (ttaad2257m000)** session. We recommend that you add this session to a job, so that the trust store is frequently updated.
   c. If a trusted timestamp is required, add an entry in the **Timestamp Providers (ttaad2558m000)** session.
   d. In the **Keys by User (ttaad2556m000)** session, add an entry for each user that must sign a document. This step is not required if documents only require seals instead of signatures.
   e. In the **Seals (ttaad2559m000)** session, add an entry for each signing key that can be used to seal documents. This step is not required if documents only require signatures instead of seals.
2. Specify the Document Output Management options for digital signing
   a. In the **Report Rules (ttrpi2520m000)** session, set the signing options on the documents that require a signature. If the **Distribute as** field is set to **`Addendum`**, and the **Separate File** check box is cleared, then signing options are not applicable. The addendum is merged into the report that is distributed as main document and that report can be signed.
   b. In the **Document Types (ttrpi2510m000)** session, define whether the document must be signed in person, or sealed. If a document must be signed, you can allow signing of multiple documents in one action.
   c. In the **Allowed Signers (ttrpi2518m000)** session, define which users are allowed to sign documents for this document type. This step is not required if the document type requires seals instead of signatures.
   d. In the **Seals (ttrpi2519m000)** session, define which key should be used to seal the document.
3. Print and sign documents
   a. Print the report that should be signed to the Document Output Management device. If a document requires a signature, then, after rendering, the status becomes **`Ready for Signing`**.
   b. In the **Documents (ttrpi3510m000)** session, complete these steps:
      1. Select a document that has status **`Ready for Signing`**.
      2. To start signing, on the *appropriate* menu, select **Sign Document**.
      3. In the dialog box that is displayed, select the parts of the document you want to sign and specify your password. After signing, the status of the document changes to **`Signed`** and the document is distributed.

         If the document type allows signing only after viewing the document, the **View Document (ttrpi3110s200)** session is started instead of the dialog box. In the **View Document (ttrpi3110s200)** session, use the Sign command to sign the document.

Chapter 29: Miscellaneous

Miscellaneous contains topics that describe a variety of information that does not belong in other sections.

Workbench applications

You can install and deploy workbench applications in LN.

The LN workbench applications run in a Web UI environment, which is hosted in an Infor Ming.le environment. The workbench applications can only be accessed within the intranet environment, not from the internet.

For details about the system requirements and the installation steps, see the *Infor LN Workbench Application Installation Guide*.

Quick Flow

Various features in LN sessions are enabled for Quick Flow.

Infor Ming.le-LN Plug-in users can use Quick Flow to optimize a specific process by suppressing process steps.

For example, to optimize print processes or process actions in a session's menu or toolbar, a user can suppress these components:

- Device selection dialogs for reports
- Option dialogs for menu actions
- Question dialogs
- Message dialogs

Note:

- To ensure users can suppress messages and questions, you must specify LN parameters and authorize the users.

  See Enabling message and question suppression on page 215.

- Quick flow is only available for form commands where the **Suppression of Dialog Allowed** check box is selected. See the online help of the **Form Commands (ttadv3118s000)** session.

The procedure, for end users, to use quick flow/message suppression is documented in these guides:

- *Infor LN UI User Guide*
- *Infor Ming.le-LN Plug-in - User Guide (Web UI)*

Enabling message and question suppression

To ensure users can suppress messages and questions:

1 Specify LN parameters. Complete these steps:
   a Start the **Tools Parameters (ttaad0100m000)** session.
   b Select these check boxes:
      - **Allow Message Suppression**
      - **Allow Question Suppression**

2 To authorize a user to suppress messages and remember question selections, you must modify the user's user data template. Complete these steps:
   a Start the **User Data Template (ttams1110m000)** session.
   b Navigate to the template you want to modify.
   c Select these check boxes:
      - **Manage Suppress Questions**
      - **Manage Suppress Messages**
   d On the *appropriate* menu, select **Convert Changes To Runtime DD**.

All users that are linked to the modified template are now authorized to suppress messages and remember question selections.

MS Excel integration

In LN sessions that show data directly from a table, you can export data to, and import data from, MS Excel.

After starting such a session, you can select the tabs and columns you want to export. Then you can perform a quick export or an advanced export. During the export, a `.xlsx` workbook is generated.

After exporting data from a session, you can edit the Excel export workbook and import the workbook back into the same session. Only cells with edit permissions are imported.

Note:
- The export functionality is available in Infor Ming.le-LN Plug-in and classic Web UI.
- The import functionality is available in Infor Ming.le-LN Plug-in.

For details, see these sections:

- "MS Excel integration" in the *Infor Ming.le-LN Plug-in - User Guide (LN UI)*

- "MS Excel integration" in the *Infor Ming.le-LN Plug-in - User Guide (Web UI)*
- "MS Excel integration" in the Web UI online help

SSI_MAX_ROWS

If a session contains much data, a full export from that session can slow down or block the entire database. To prevent this, you can limit the number of records that can be exported by setting the SSI_MAX_ROWS parameter.

To limit the number of records that can be exported, set SSI_MAX_ROWS to a value greater than zero. For example, if SSI_MAX_ROWS = 500, 500 records can be exported at most.

If SSI_MAX_ROWS has a value less than or equal to zero, there is no limit, except the MS Excel limit, for the number of records that can be exported.

System Messages

To send system information to users and administrators, you can use system messages. The **System Messages (ttaad3290m000)** session is created to send messages to specific users and administrators on the server.

To run this session in an on-premises environment, you must be logged on as a super user. The first tab only is available to edit. The message that is created here is shown to all users on the server.

In a multitenant environment several tabs are displayed and the landlord can specify information in all tabs. The super user of the tenant can only specify information in the first tab.

These tabs are displayed to the administrator in a multitenant environment:

- Own Users
- All Tenant Admins
- Single Tenant Admins
- All Tenant Users
- Node Users

Every tab has its own form commands, such as:

- Save All
- Save
- Revert to Saved
- Preview

This session reads and writes the $BSE/lib/systemmess file for messages to the Own Users.

Glossary

3GL script

A program script that can be linked to sessions without forms, or that is not linked to a session at all. 4GL statements and sections cannot be used in 3GL scripts. In other words the entire program flow, including the main function, needs to be specified.

4GL engine

The program that provides default functionality for a session to prevent application programmers from having to develop a session from scratch. The 4GL engine, formerly called standard program (STP), is used because essentially sessions are alike. In addition to providing default functionality, the 4GL engine also provides a mechanism to change the 4GL engine's default behavior, and to program dedicated functionality for a specific session. Each time a session is started, a separate 4GL engine instance is activated to handle the session.

4GL script

An event-oriented program script that is linked to a session. The instructions can be specified in program sections, form sections, field sections, main table input/output sections, choice sections, zoom from sections, and functions.

Active Directory Service

Active Directory is an implementation of LDAP directory services by Microsoft for use in Windows environments. Active Directory allows administrators to assign enterprise wide policies, deploy programs to many computers, and apply critical updates to an entire organization. An Active Directory stores information and settings relating to an organization in a central, organized, accessible database. Active Directory networks can vary from a small installation with a few hundred objects, to a large installation with millions of objects.

additional file

A generic component, such as an XML schema file, a GIF image, a JPG image, a PNG image, or an SVG image. From LN 6.0a on, additional files are stored in a specific package, module, and package VRC.

algorithm

A detailed finite sequence of actions that must be performed to accomplish some task.

Application Foundation Classes

Microsoft's Application Foundation Classes, a collection of Java classes for creating full-fledged application programs. AFC is a comprehensive set of Java class libraries providing user-interface controls, graphics and effects classes, and the ability to create and extract cabinet (CAB) files.

application help

Help that tells you how the application works. An online manual contains an overview of the packages and the modules in each package and explains the functionality of each module in detail, describing the underlying concepts and applied methods.

Application Programming Interface

A set of methods that can be invoked by other applications. An application's API enables other programs to retrieve data or to carry out functionality from that application.

Application Services Manager

A stand-alone product that can start and stop LN application services.

appropriate menu

Commands are distributed across the **Views**, **References**, and **Actions** menus, or displayed as buttons. In previous LN and Web UI releases, these commands are located in the **Specific** menu.

array

A list of data items of the same type with the same name. An element in the array can be referenced by an expression composed of the array name and an index expression. Multilevel arrays are used for data storage in tables.

asynchronous communication

Communication between processes, where the process that sends the message does not expect, or wait, for a reply from the receiving process.

attachments

Allows you to create or view documents that are attached to the selected record. If you use object data management, you can attach documents to LN records such as items, production orders, and service orders.

attribute

Literally, a characteristic. In databases, the name or structure of a field is considered an attribute of a record. Structure refers to its size and datatype. For example, LASTNAME, 50 characters, text. In screen displays, additional information stored with each character in character mode that controls the background and foreground colors of the character, blinking, and underlining.

audit

To create an audit trail that traces all activities that affect a piece of information, such as a data record, from the time it is entered into the LN database to the time it is removed.

audit category

A group of audit profiles, used to group audit profiles of the same functional area.

audit data dictionary

A representation of the definition of an audited table in the data dictionary at a certain point of time. When a sequence file is created, the audit DD is kept in the sequence file's header. When the data dictionary changes in a way that affects the table, a new sequence file is created and the new audit DD is added.

audit host

The system where the audit server runs.

audit management

A management system that controls the auditing process. Audit management can be used, for example, to specify for which database tables an audit trail must be created and where the audit trail's data must be stored.

audit profile

A group of audit settings, configured for specific companies, tables and fields.

audit server

A program that creates the audit trail.

audit trail

A means of tracing all activities affecting information, such as data records in a database, for a defined time interval. The information is stored in the audit files.

audit type

The setting that defines when a field is audited.

You can choose to audit a field 'always', or only when it is 'changed':

- If a field is audited 'always', the field is logged each time when the content of the field, or the content of any other audited field, changes.
- If a field is audited when it is 'changed', the field is logged only when the content of the field itself changes.

Authorization Management System

An authorization system in LN that defines the permissions of a normal user. The user is assigned to a group of users with the same role and/or sub role in an organization. LN links the authorizations to the role of the user group, not to the individual users in the group.

authorizations

A set of permissions that limit the access to various objects in LN. For example, access can be denied to sessions, tables, and companies.

base language

The fallback language that is used to display application data.

LN uses the following fallback mechanism:

1. LN displays the application data in the data language that is specified in your user data.
2. If no data language is specified in your user data, LN displays the application data in the data language that is linked to your software language.
3. If no data language is linked to your software language, LN displays the application data in the base language.

base VRC

A means in PMC to identify products in a unique way. Updates at the distributor side are provided with the base VRC identifier. A base VRC can contain the code of the physical VRC in which the related master product is installed, for example, B61_a, but can also be a code not related to a physical VRC, for example, 7.6_a_tt. At the recipient side, every update VRC is linked to a base VRC identifier. The installation process checks if the base VRC identifier of the update matches with the base VRC identifier of the update VRC. If not, you cannot install the update in that update VRC.

base VRC combination

A *Base VRC* combination is defined at the PMC distributor side and consists of a set of related base VRCs. A base VRC combination controls the creation of co-requisites between base VRCs. You can only define co-requisites between base VRCs that are part of the same base VRC combination. Base VRC combinations prevent the unwanted creation of co-requisites between base VRCs.

batch-oriented replication

A time-driven data replication method, as opposed to the real-time data replication. The batch-oriented replication process is started based on a schedule, for example, every hour or every week.

bi-directional language support

A function that is necessary to support languages that are read and written from right to left. For example, Arabic, and Hebrew.

blocking process

A process that must be completed before other processes can continue.

BSE path

An environment variable containing the directory where the LN software is installed.

business object

A business object is an object understandable by the business, such as a purchase order or an organizational unit. A business object has information stored in the business object attributes, such as the purchase order number or the organizational unit name. A business object also contains a set of actions, known as business object methods, that can manipulate the business object attributes, such as create purchase order and list organizational units.

From a development perspective, a business object is a collection of tables and functions that manipulate these tables that are implemented simultaneously as one group during the development phase. A business object is identified by the combination of a package code, module code, and business object code.

Business Object Interface

Business Object Interfaces provide a connection between partner applications and third-party applications and the LN software, as well as connecting LN functional components. Business Object Interfaces are developed for situations where the LN software acts principally as a server, and a client software invokes the methods in the objects.

Business Object Layer

A set of standard DLLs from LN 6.0 onwards. The Business Object Layer invokes LN business logic methods that handle the persistency of a Business Object

business object meta data

Describes the structure and content of a Business Object and its mapping to LN tables and DLLs.

Business Object Repository

A persistency system that stores the Business Object meta data. The BOR is a virtual space in which Business Objects are modeled. The runtime objects of the business objects are stored in the Business Object Layer.

business partner

A party with whom you carry out business transactions, for example, a customer or a supplier. You can also define departments within your organization that act as customers or suppliers to your own department as business partners.

The business partner definition includes:

- The organization's name and main address.
- The language and currency used.
- Taxation and legal identification data.

You address the business partner in the person of the business partner's contact. The business partner's status determines if you can carry out transactions. The transactions type (sales orders, invoices, payments, shipments) is defined by the business partner's role.

C

A structured programming language. C is a compiled language that contains a small set of built-in functions that are machine dependent. The rest of the C functions are machine independent and are contained in libraries that can be accessed from C programs. C programs are composed of one or more functions defined by the programmer.

character set

The smallest component of a written language that has semantic value (that is, linguistic meaning). A character set is the collection of characters that is used in a given language. As well as the alphabet, a character set includes numerals, mathematical and currency symbols and punctuation marks. The character set that you select determines how texts are displayed and printed by your computer, for example as Latin or Arabic characters. Examples of character sets include ASCII and ISO 8859-1.

chart

A graphic or diagram that displays data or the relationships between sets of data in pictorial rather than numeric form. The data can be presented in a graph, a line, or a pie, and can include titles, legends, and footnotes.

chart application

A program that is used to send data from a LN session to the Chart Manager. A chart application is linked to a package VRC to customize the attributes that are specified in the chart.

chart type

The chart type determines what the chart looks like.

For example, it defines the type of graph, the thickness of lines, the size of bars, and the colors. The following default chart types are present in LN:

- Bar
- Layer
- Line
- Pie
- Scatter
- Stacked bar

check in

A process that releases the checked-out software component and copies the software component from the Development VRC to the Original VRC. This process also stores a historical version.

A revision text for the changed software component is required for a check-in process.

check out

A process that locks the software component for other developers. During the check-out phase, other related components are locked as well while the component can be updated and tested. A copy of the component is made from the original VRC to the *Development VRC*.

child field

A table field that is part of a combined field. A combined field contains two or more table fields, which are the child fields of the same table field.

class

In object oriented programming, a class is a generalized category that describes a group of more specific items, called objects, that can exist within it. A class is a template definition of the methods and properties (variables) in a particular kind of object. Thus, an object is a specific instance of a class that contains real values instead of variables. The class is one of the defining ideas of object-oriented programming. Among the important ideas about classes are: A class can have subclasses that can inherit all or some of the characteristics of the class. In relation to each subclass, the class becomes the superclass. Subclasses can also define their own methods and variables that are not part of their superclass. The structure of a class and its subclasses is called the class hierarchy

client

A user, program or system that requests the execution of a specific task from another program or system. See also *server*.

client/server environment

A distributed system in which software is split between server tasks and client tasks. A client sends requests to a server, according to a protocol, asking for information or action, and the server responds.

collection

In PMC, a collection is a group of individual solutions. At the PMC distributor side, you can perform grouping in various ways, for example, manual grouping based on a functional topic or grouping based on solutions created in a particular period etc. You cannot define dependencies between collections. At the recipient side, the entity collection is not available. When a collection is scanned, the individual solutions are added to the PMC registry and can be processed individually.

combined field

A combination of two or more child fields of the same Table.

commercial product

A set of tools, directories, or sessions which you can deliver to a customer.

commit

An action to successfully complete a transaction in order to make the database changes permanent.

commit protocol

An action to successfully complete a transaction in order to make the database changes permanent.

common user data

The parts of the user data that are similar for a group of users. The authorization data is not included in this definition. The user authorizations are defined in the Authorization Management System.

company

A means to store data separately in LN. Every company has its own tables with its own data.

Example: An Item with Item Code "X" can be stored in the Item Master Table of several different companies. In each company this Item Code "X" can have different data.

Note: Company number 000 is reserved by Infor to store the Enterprise Server (Tools) data.

component management

A system that a developer uses to lock software components, that will be changed, for other users.

co-requisite

In general, co-requisites are defined between solutions of a standard product and derived products. Co-requisites guarantee that related products are updated simultaneously under the condition that the update VRCs of the related products are linked to the same VRC combination. The order of installation is not relevant. The solutions can have the same *base VRC*, or different base VRCs.

customer

A code that identifies a customized software package. For example, B61O_a_cus1, where 'B61O' represents the version, 'a' represents the release, and 'cus1' represents the customer extension of the customized software.

customization

A derived product, meant to fulfill the requirements of a specific customer or group of customers.

daemon

A program associated with the UNIX operating system that permanently runs in the background. A daemon handles requests that are issued by other programs and is activated only when needed. For example, the job daemon runs permanently on the LN server. If the job daemon detects that a job is present, it executes the job at the job's scheduled time.

Data Access Layer

A Dynamic Link Library that is linked to a LN database table.

database

A collection of data that is organized so that its contents can easily be accessed, managed, and updated. A database has tools to search for data in the database, sort the data in the database, and so on.

database administrator

The individual responsible for the installation, planning, configuration, management control, and maintenance of the relational database.

database authorizations

The set of sessions that can be used to print, display, maintain and convert the authorizations that a group of users with the same role, has for specific tables and table fields in the database.

database group

A group of database users with the same authorizations and rights. When you add a database user to a database group, the user automatically receives the group's authorizations and rights.

Database Management System

A software interface between the database and the application software. A database management system handles user requests for database actions and allows for control of security and data integrity requirements.

database user

An LN user that is also authorized to access the database. The configuration information for the database user contains all the necessary settings to log on to a database. When the user logs on, these settings are automatically loaded. Every database user must belong to a database group.

Data Dictionary

A database containing data about all the databases in a database system. Data dictionaries store all the various schema and file specifications and their locations. They also contain information about which programs use which data and which users are interested in which reports.

Data Director

The LN package that contains the Exchange module.

data label

Contains the name or the value of a data point.

data language

The language in which the application data is displayed. Data languages can be linked to users and to software languages.

A data language consists of an ISO 639-1 language code and, optionally, an ISO 3166-1 country code.

data point

A point in a chart that represents a value and is made visible by means of a symbol, bar, pie segment, or a step in a line. The data point value is determined by the chart series, the category, and the data value.

data replication

A method of table sharing in which every modification of a table is copied to the companies that share the table.

In LN, you can perform Batch-oriented replication by using the Exchange module.

Data Upgrade Engine

A framework that executes application data *upgrade programs*.

The Data Upgrade Engine is used after installation of a *Feature Pack*.

data upgrade run

The execution of multiple *upgrade programs* for multiple *companies*.

dependency

In PMC, the relation between solutions. Dependencies are defined at the PMC distributor side and are part of the meta data of a PMC solution and guarantee that PMC solutions are installed in the correct configuration and sequence at the PMC recipient side.

The following values indicate the dependency type between solutions.

Three dependency types are available:

- *Pre-requisites*
- *Co-requisites*
- *Post-requisites*

You can only install solutions that are dependent on other solutions if the other solutions are already present, or are also installed.

The same dependency types exist between *patches*. However, to keep the descriptions readable, only solutions are mentioned, but patches are meant as well. One exception applies: the post-requisite type is not applicable to patches.

derived product

A product that is derived from a parent master product. The parent product can, for example, be a main LN version such as LN 6.1. In general, the derived product contains a subset of components from the parent product that are changed and possibly a number of new software components.

There are three categories of derived products:

- *Localizations*
- *Extensions*
- *Customizations*

details session

A dialog box that shows all the details (fields) of the line (record) selected in the associated overview session. Use a details session to view, enter, or change the data of one record.

A details session can contain a number of tabs to group related fields.

development VRC

In PMC a physical VRC, derived from the *Export VRC*, in which checked-out software components are temporarily stored during a change process.

device

A generic term for a computer subsystem. For example, a printer, serial port, or a disk drive.

domain

A domain describes the properties of table fields. The main property of a field is the data type, e.g. string, long, double, date, UTC. Other properties are, for example, the length of a string, the alignment of a string, and the date and time format.

In the Chart Manager, a range with an initial value, an end value, and a step size. Domains indicate the valid values for an attribute and are used to define the scale division of the axes or to verify data.

driver

A hardware device or program that controls other devices. A printer driver, for example, is designed to enable other programs to work with a specific printer regardless of the printer's hardware specifications or language.

dump

To copy data to an external source.

dynamic form

A form with a dynamic form definition.

The developer does not need to determine exactly where fields must be placed, or what they must look like. Instead, the developer defines:

- The form contents.
- The form structure.
- The sequence of the objects on the form.

At runtime, the dynamic form displays only those fields for which the user is authorized.

Dynamic Link Library

A means of sharing common functions between several programs. This library contains functions for common use. The library can be linked to the object as a function call at run time. Implementation of a dynamic link library reduces the size of objects to a minimum because dynamic link libraries are not included in a programs object.

dynamic session

A session with a dynamic form definition.

Depending on specific settings, a dynamic session can start in these ways:

- As a details session or as an overview session.
- As a details session only.
- As an overview session only.

In the dynamic form definition, the developer does not need to determine exactly where fields must be placed, or what they must look like. Instead, the developer defines the form contents, the form structure, and the sequence of the objects on the form.

At runtime the dynamic form displays only those fields for which the user is authorized.

element

A general term used for entities defined in LN, such as items, business partners, currencies, and so on. The details of an element are registered in the database as a record. In an overview session each element or record is represented by a line.

eMessage connector

An interface between LN and various messaging providers, which is based on XML and independent of the message protocol. For example, fax, telex, e-mail, SITA, or SMS.

encryption

The process of encoding data to prevent unauthorized access, especially during the transmission of data. You usually need a key to decode encrypted data.

expired

If a software component, such as a session or a table, is set to expired in a certain VRC, it cannot be used anymore in that VRC and in later VRCs which were derived from the VRC where the component was set to expired.

A software component that was set to expired is still available in preceding VRCs.

export VRC

The physical VRC from which components that belong to a PMC solution must be exported at the PMC distributor side. Each *base VRC* has an export VRC linked, so components for different products are exported from different physical VRCs.

expression

A combination of symbols that produces a result upon evaluation. For example, identifiers, values, and operators. The resulting value can then be assigned to a variable, passed as an argument, tested in a control statement, or used in another expression.

extension

A derived product meant to fulfill requirements of a particular line of business.

Feature Pack

See *Service Pack*.

field

In table definitions, a field refers to a column. In a session, a field is a specified area of a record used for a particular category of data.

field help

Help that explains the field's value and how it is used in the session. Field help also mentions the default values and which values are possible or allowed. Related topics point you to the related functionality in the online manual.

filter

A means to selectively display specific information.

In LN, two types of filters exist:

- A way in which LN users can display and sort a specific subset of records in an overview session.
- A program that suppresses the display of specific fields in a session, depending on the user's authorizations. These form field authorization filters are used during the form definition. The filters are determined by the roles that you specify in the Authorization Management System.

float

A data type name used to declare variables that can store floating-point numbers, that is, any number containing a decimal point, with a maximum of 15 significant digits (8 bytes).

font

A set of characters of the same typeface, style, and weight. For example Arial, in italic, and bold.

form

A screen that appears when a session is started. A form interacts with the database, and provides the user interface that is used to manipulate the data on the form.

form command

A command:

- that starts a session, function or (sub)menu by means of which a user can carry out a particular task.
- that, as opposed to standard menu commands such as the **Exit** command, must be especially defined for a session tab.

form field

A field that is shown on a form. A form field is selected from the available fields of an input table and its reference tables.

function

A piece of program code that makes up part of a program script.

A function is a self-contained software routine that can perform a task for the program in which the function is written, or for another program.

glossary

Definitions of the terms used in LN.

graph

A graphical representation of information in a coordinate system. Quantities are represented by bars, lines, symbols, or combinations of these.

graph series

A number of data points that are linked graphically by means of a line, a number of symbols, bars, or combinations of these.

group

A set of form objects grouped together. Examples of form objects are form fields and child groups.

group-by field

A field on an overview form that is positioned above the grid. The group-by field determines what is shown in the grid of an overview form. Only the records that belong to the group-by fields are shown in the grid, for example, all orders that belong to a specific customer. The name of the customer is shown in the group-by field, the records are shown in the grid.

help components

Several sorts of online help used in LN.

The help components in LN are:

- Session help

  Describes the general purpose of a session and how the session must be used.
- Field help

  Describes the meaning of the field and helps you to enter a correct value in the field.
- Domain help

  Describes the option list values of a field.
- Command help

  Describes the command buttons, toolbar buttons, and menu commands.
- Glossary

  Describes a definition of terms that are frequently used in LN.
- Online manuals

  Describes the concepts and methods that are used in LN.

help topic

The code of the help text that is used to link help texts to software components. Each text has a help text code and belongs to a package VRC.

index

One or more table fields used to sort and search records in a *table*. A table must have at least one index. The first index is always the Primary key.

installation run

In PMC, a group of solutions that were installed together. This can be a range of solutions, a solution with *pre-requisites*, or a combination of both.

integrated session

The session and the session's form are integrated into one object. The form is a subcomponent of the session.

When you perform an operation, for example, copy, delete, check-in, or checkout, on an integrated session, you also perform the operation on the integrated session's form.

A non-integrated session's form is a separate object.

job

One or more sessions and/or shell scripts that LN executes without user interaction. You can schedule jobs to run periodically or at a specified time.

job calendar

A calendar that consists of a number of dates/times, which are used to indicate the dates/time a periodical job must be executed.

label

A code that is used instead of language-dependent text in forms, reports, and menus. A label consists of a name and a content description. The content of a label can differ by language, but the label name remains the same for all languages.

language number

A conversion of the language code to a number between 0 and 61. The language number eliminates problems caused by the use of uppercase and lowercase language codes. The language code corresponds to the language number by the following convention: Language code range Corresponding language number range 0 to 9 0 to 9 a to z 10 to 35 A to Z 36 to 61.

Example

Language code b = Language number 11 Language code B = Language number 37

library

A collection of files, computer programs, or subroutines.

library authorizations

The set of sessions that can be used to print, display, maintain, and convert the authorizations of a group with the same role has for specific Dynamic Link Libraries.

license

A means to validate the system configuration request of the customer.

Lightweight Directory Access Protocol

In computer networking, the Lightweight Directory Access Protocol, or LDAP, is a standardized networking protocol designed for querying and modifying directory services.

LN 3GL

A third-generation proprietary programming language that is a mix of Basic and C.

LN 4GL

A fourth-generation language is a programming language designed for interacting with the programmer used with relational databases. 4GLs are event-driven.

LN BW Environment and Configuration Selector

A tool that makes it possible to run different BW versions on a single client.

LN shell

The LN shell is a program that runs the LN applications. The LN shell serves as an interface program between the LN applications, operating system, user interface, and database. This open systems architecture allows the LN applications to operate on all supported combinations of operating systems, user interfaces, and databases.

LN software environment

The directory structure on the operating system that contains the LN software and data dictionary.

LN user

A person who uses the LN application.

LN windows interface

A Windows-based interface program between the user and the LN servers. The LN windows interface runs on the user workstation and communicates with the Bshell.

locale

The character set and language used by a system.

localization

A derived product meant to fulfill country specific requirements.

log

verb: To create a record of transactions or activities that take place on a computer system. noun: A record of transactions or activities that take place on a computer system.

logical company

Company as seen by the application. The sessions and program scripts of LN use the logical companies without having to know the physical company where the tables are actually stored.

In the most simple situation, a table in a logical company is linked to a physical company with the same company number as the logical company. In that case, the logical company and the physical company are the same company. If you use logical logical table linking, a table in a logical company is linked to a physical company with a different company number.

logical printer

A logical printer is a set of printer devices. In some environments several printers may be located in close proximity to one another (same room/floor). They can be grouped together as a logical printer. When output is directed to a logical printer, the printer daemon will check the paper type of each printer. It will automatically direct output to the first printer that contains the required paper type.

long

A data type specified in LN as any whole number from -2147483648 to 2147483647.

menu

A list of options from which a user can make a selection to perform a desired action, such as starting sessions, other menus, and queries. A start-up menu is defined for each user. Using this start-up menu, the user can access all sub-menus attached to the start-up menu tree.

merge

A process which creates a target VRC from a source VRC.

message

A notification that informs you about something. More specifically, a message attends you to an event, error, warning, and so on. Messages usually appear in a message window or logged in a file. If displayed in a window, a message requires a confirmation: Click **OK**. Messages are distinguished from *Questions*, which always require a choice response.

message server

A server that stores and sends messages to clients

module

A part of a package consisting of a number of related software components, such as sessions, tables, program scripts, reports, forms and menus. For example, the General Ledger module in Financials.

A module code consists of three characters. For example, the General Ledger has the code "gld".

multi occurrence

A session that lists the available elements or records of one type, and some of their details (fields). You can use an overview session to view, sort, add, change, copy, and remove records.

When you add or change a record a details session usually starts. In some cases, you can add and change records directly using the overview session.

Native Language Support

Native Language Support. The system to convert character sets between different computer systems. In LN it normally refers to conversion between the character set of the operating system (for example Windows NT or UNIX) and the bshell. The native language is the character set which is used by the operating system on which LN is running.

Network Information Service

The Network Information Service or NIS is Sun Microsystems' "Yellow Pages" (YP) client-server directory service protocol for distributing system configuration data such as user and host names between computers on a computer network. NIS/YP is used for keeping a central directory of users, hostnames and most other useful things in a computer network. For example, in a common UNIX environment, the list of users (for authentication) is placed in /etc/passwd. Using NIS adds another "global" user list which is used for authenticating users on any host.

object model

A set of defined object types. The model defines which object types support Document Authorization.

object type

Defines a business object, or business document, such as a sales order or a contract, in the context of Document Authorization.

An object type consists of one or more tables, and specifies this information:

- How the tables of the object type relate to each other.
- The actions to generate a request for Document Authorization, which will be processed in ION Workflow.
- What data of the object type must be sent to ION Workflow to perform the actual Document Authorization.

obsolete solution

Obsolete solutions are an administrative aid to manage the synchronization of updates at the PMC recipient side when you install a *Service Pack*. An obsolete solution does not contain software components.

operator

A symbol or other character indicating an operation that acts on one or more elements.

original VRC

The VRC that contains the software components that have to be modified. These software components will be changed in the Development VRC.

overview session

A session that lists the available elements or records of one type, and some of their details (fields). You can use an overview session to view, sort, add, change, copy, and remove records.

When you add or change a record a details session usually starts. In some cases, you can add and change records directly using the overview session.

package

A set of related modules that implements a complete part of the functionality, for example, Enterprise Planning, Financials, or Warehousing. Packages are designed to function as independent as possible, to enable a customer to implement only particular packages.

A package code consists of two characters. For example, tt is the code of the Tools package.

Each package has a unique version structure, the VRC structure.

package combination

A combination of several different packages with specific VRCs. A package combination represents a complete usable version of LN.

In the User Data (ttaad2500m000) session, each user is linked to a package combination, that determines which version of the software the user can use. In the Companies (ttaad1100m000) session, each company is linked to a package combination, to indicate which version of LN is appropriate to handle the data in that company.

package VRC

A version of a package, for example, **tc B61O a cus1**. In general, one version of a software component, such as a session, a table, or a form, is stored in one particular package VRC.

A developer can usually modify software components only in a particular package VRC.

The code of a package VRC consists of:

- Package code, for example, 'tc'
- A version (VRC) code, for example 'B61O a cus1', built up of:
  - Version
  - Release
  - Customer

paper type

A particular type of paper that must be used for a printer. The paper type must correspond with the paper type in the printer.

password

A password is a form of secret authentication data that is used to control access to a resource. The password is kept secret from those not allowed access, and those wishing to gain access are tested on whether or not they know the password and are granted or denied access accordingly.

password aging

Password aging is a technique used by system administrators to defend against bad passwords within an organization. Password aging means that after a set amount of time, usually 90 days, the user will be prompted to come up with a new password. The theory behind this is that if a user is forced to change his password periodically, a cracked password is only useful to an intruder for a limited amount of time. The downside to password aging, however, is that users are more likely to write their passwords down. With password aging the system administrator can set minimum and maximum lengths of time for which the password is valid. Maximum time lengths force users to change passwords regularly. Minimum lengths prevent users from quickly changing the passwords back.

patch

In PMC, a patch is a collection of *Solutions*. In general a patch contains solutions created in a larger time period. The patch entity is both known at the *PMC distributor* and *PMC recipient* side. Patches are an indivisible set of solutions. You cannot install or uninstall individual solutions that belong to a patch at the PMC recipient. You can only install or uninstall patches as a whole. Yo can define dependencies between patches. Patches leave the *Base VRC* that is linked to the *update VRC* at the PMC recipient unchanged. The existing PMC registry will remain and will be extended with data of the newly installed patch. Patches only permit the most recent version of software components to be maintained. Patches in general mainly contain corrective solutions.

Note: In *PMC versions* earlier than LN 6.1, the synonym Service Packs was often used for patches.

periodical job

A job that can be executed more than once. For a periodical job, you must specify a time interval or calendar. If desired, the job can be automatically reactivated.

physical company

The company where the data is actually stored. You can link tables in *Logical companies* to the corresponding tables of a physical company.

pie segment

Represents a percentage of the total pie chart.

Pluggable Authentication Modules

Pluggable authentication modules or PAM are a mechanism to integrate multiple low-level authentication schemes into a high-level API, which allows for programs that rely on authentication to be written independently of the underlying authentication scheme. PAM were first developed by Sun Microsystems, and are currently supported in AIX, HP-UX, Solaris, Linux, FreeBSD and NetBSD.

PMC distributor

The functional part of PMC that manages the creation of *Updates*. PMC Distributor is especially used by software vendors who create updates.

PMC recipient

The functional part of PMC that manages the installation of *Updates*. Customers, who install updates in particular use PMC recipient.

PMC version

PMC uses a version number in the solution dumps to ensure that the PMC software at the recipient side can process these dumps. If the PMC version at the recipient side is too low to process a particular dump, then an error message is displayed. In this error message, a solution number is displayed. You must first install this solution, before you can process the original solution dump.

This mechanism enables you to change the format of the PMC dumps when you must include new functionality.

The PMC version is linked to every update. This guarantees that various formats of PMC solution dumps are handled by the correct version of the PMC software. After installing solution 2177498, the PMC version of the client can be found in the Parameters (ttpmc0100s000) session.

Note: Dumps created for a higher PMC version cannot be processed at the recipient side if the recipient has not been upgraded to that PMC version. Dumps of lower PMC versions can always be processed.

The PMC version to be used at the distributor side depends on these factors:

- The Enterprise Server version (Infor LN Tools version) at the distributor side
- The functionality you want to use in PMC
- The Enterprise Server version of the customers for which you are creating PMC solutions

You can edit the PMC version at the distributor side in the session.

porting set

A collection of objects such as bshell, database, drivers, printer daemon and so on, which can only run on the machines/operating systems for which they are compiled/ported.

post-requisite

Post-requisites are mainly meant to prevent the installation of bad solutions. In general, a post-requisite is a link from an earlier, bad solution to a more recent, correct solution.

pre-requisite

Pre-requisites mainly steer the sequence in which solutions are installed. In general a pre-requisite is the link from a more recent solution to a predecessing solution.
Pre-requisites are the most common type of dependencies. A pre-requisite dependency exists between two solutions if one solution must have been installed before the other solution is installed. In that case, the first solution is a pre-requisite for the other solution. Typically, pre-requisite dependencies exist between a solution and a previous solution, if these solutions have one or more components in common. Pre-requisite dependencies can only be created to solutions in the same *Base VRC*.

primary key

The unique identification for a record in a Table.

printer daemon

A networking program that permanently runs in the background and handles the print requests issued by the users. When output is directed to a logical printer, the printer daemon checks the paper type of each printer, and then directs the output automatically to the first printer that contains the required paper type.

Product Maintenance and Control

Product Maintenance and Control (PMC) is a tool that helps a customer manage the updates of the LN system.

With the PMC tool, you can check all patches against the customer's LN system to verify their completeness, check any potential interference with the customization, and detect dependencies.

These capabilities ensure the complete and accurate installation of each software patch and Service Pack. In addition, using the PMC tool enhances the quality of the support being available to customers.

PMC consists of a *PMC distributor* part and a *PMC recipient* part.

program script

A sequence of instructions that is used to program a number of actions that must take place in addition to the standard program. Two different program scripts are available, namely 3GL scripts and 4GL scripts.

query

The process of extracting information from a database and presenting it in a report.

question

A notification that requires a choice response. For example, a question can prompt you to confirm or cancel a delete action. If you do not respond to a question, the process that prompted the question cannot continue. Questions are distinguished from Messages, which offer no choice and do not necessarily require a response.

question mark button

Changes the pointer into a question mark. To display pop-up Help that briefly describes the command, click a menu command, a command button, or a toolbar button with the question mark pointer.

To display the field Help, click a field with the question mark pointer.

record

A data structure that is a collection of fields (elements), each with its own name and type. The elements of a record represent different types of information and are accessed by name. A record can be accessed as a collective unit of elements, or the elements can be accessed individually.

reference mode

The way in which a reference restricts the contents of a table field.

A reference can have one of the following reference modes:

- **`Mandatory`**

  The field must contain a code that is found the reference table.
- **`Mandatory unless empty`**

  The field can be empty. If the field is not empty, the field must contain a code that is found the reference table.
- **`Not mandatory`**

  The field can be filled with a code that is not found in the reference table. The reference only serves to expedite queries.

reference table

The table to which some table field refers.

Example

One of the fields of the **Items – General** table is the **Country of Origin** (coor) field. This field can contain a country code. (The field can also be left empty.) LN stores country codes in the **Countries** table. To control this connection, the table field **Country of Origin** in the **Items – General** table has a reference to the **Countries** table.

Items – General is the referral table and **Countries** is the reference table.

reference type

The means by which a reference is defined.

- **`Hard`**

  If the reference type is **`Hard`**, the reference is defined in the table definition. This reference type is the most common.
- **`Soft`**

  If the reference type is **`Soft`**, the reference is not defined in the table definition, but programmed in a program script or dll.

referral table

The table that has a field that refers to another table.

Example

One of the fields of the **Items – General** table is the **Country of Origin** (coor) field. This field can contain a country code. (The field can also be left empty.) LN stores country codes in the **Countries** table. To control this connection, the table field **Country of Origin** in the **Items – General** table has a reference to the **Countries** table.

Items – General is the referral table and **Countries** is the reference table.

Relational Database Management System

A relational database or database management system that consists of tables made up of rows and columns. In a relational database, the rows of a table represent records and the columns represent fields with the particular attributes of a record. The records are collections of information about separate items. The attributes of the records are represented in the columns. The relational database management system uses data in a specified column of one table to find additional data in another table.

release

An update in a version. For example, 'B61 a', where 'B61' represents the version code, and the 'a' represents the release code. Each update results in a new release with a new release code.

report

A report is used to present data from the database, usually on paper. The report can be sent to a device, such as a physical printer, a display, or to a file.

Revision Control System

A tool, that is used by Tools to store revisions of scripts, libraries, includes and report scripts.

role

From a user point of view, a function, or part of a function in an organization. For example, manager, secretary, and so on. From an authorization point of view, an identifying name for a group of users. A role can contain several sub roles.

SCM group

A Software Configuration Management group in LN that identifies a development group that has a separate development environment.

Security Support Provider Interface

Security Support Provider Interface (SSPI) allows an application to use various security models available on a computer or network without changing the interface to the security system. SSPI does not establish logon credentials because that is generally a privileged operation handled by the operating system. A security support provider (SSP) is contained in a dynamic-link library (DLL) that implements SSPI by making one or more security packages available to applications. Each security package provides mappings between the SSPI function calls of an application and the functions of an actual security model. Security packages support security protocols such as Kerberos authentication and LAN Manager.

sequence file

The file in which (a part of) the audit trail is stored.

server

A program or system that performs a predefined task at request of a user or another program or system. See also *client*.

Service Pack

In PMC, a Service Pack is a collection of solutions. In general, a Service Pack contains solutions created in a larger time period. In PMC the term 'patch' is also applied for Service Packs. The patch entity is both known at the PMC distributor and PMC recipient side. A property in the patch entity makes the difference between patches and Service Packs. Service Packs are an indivisible set of solutions. You cannot install or uninstall solutions that belong to a Service Pack at the PMC recipient. You can only install or uninstall Service Packs as a whole. You can define dependencies between Service Packs. Service Packs are intended to enable you to maintain multiple *Base VRCs* in parallel. Service Packs change the base VRC that is linked to the update VRC at the PMC recipient. The existing PMC registry for the update VRC will be moved to history and a new registry will be started for the update VRC. This type of patch in general contains a significant amount of functional changes.

Note: Service Packs as described in the preceding definition do not exist in PMC versions earlier than LN 6.1.

service provider

The software application that enables messaging, such as Outlook or Fenestrae.

session

An elementary part of LN the user can start to run an application's functionality. Usually, a session is linked to a main database table and a program script. In addition, a session uses zero or more forms, reports, and charts.

The code of a session consists of a package code, a module code, four digits that indicate the main table number and the session type, an m or an s, and three additional digits, for example, **Countries (tcmcs0510m000)**.

session authorizations

The set of sessions that can be used to print, display, maintain and convert the authorizations that a group of users, that are identified by a role, have for specific sessions.

session group

A group of sessions that are automatically started when a user starts the bshell.

session help

Help that tells you how to use the session and what the result will be. Notes and remarks draw your attention to any special information. Related topics point you to a description of the related functionality in the online manual.

shared memory

A part of physical memory intended for common use. Programs communicate with each other through shared memory. The use of shared memory results in faster access to the components loaded in shared memory. Before starting the application, shared memory must be initialized and setup. A prerequisite for using shared memory is that it should be supported by the hardware besides having sufficient internal memory at its disposal.

The following components can be loaded into shared memory:

- report objects
- program objects
- table definitions

shell command

The program, at operating system level, which is executed before the output is (re) written to a file. For example, this program can be a conversion program to filter the input data.

single occurrence

A dialog box that shows all the details (fields) of the line (record) selected in the associated overview session. Use a details session to view, enter, or change the data of one record.

A details session can contain a number of tabs to group related fields.

Single sign-on

Single sign-on (SSO) is a specialized form of software authentication that enables a user to authenticate once and gain access to the resources of multiple software systems.

There are at least five major types of SSO or reduced sign-on systems in common use at the time of this writing (2005):

- Enterprise single sign-on (E-SSO)

  E-SSO, also called legacy single sign-on, after primary user authentication, intercepts login prompts presented by secondary applications, and automatically fills in fields such as a login ID or password. E-SSO systems allow for interoperability with applications that are unable to externalize user authentication, essentially through "screen scraping."
- Web single sign-on (Web-SSO)

  Web-SSO, also called Web access management (Web-AM), works strictly with applications and resources accessed with a web browser. Access to web resources is intercepted, either using a web proxy server or by installing a component on each targeted web server. Unauthenticated users who attempt to access a resource are diverted to an authentication service, and returned only after a successful sign-on. Cookies are most often used to track user authentication state, and the Web-SSO infrastructure extracts user identification information from these cookies, passing it into each web resource.
- Kerberos

  Kerberos is a popular mechanism for applications to externalize authentication entirely. Users sign into the Kerberos server, and are issued a ticket, which their client software presents to servers that they attempt to access. Kerberos is available on UNIX, Windows and mainframe platforms, but requires extensive modification of client/server application code, and is consequently not used by many legacy applications.
- Federation

  This is a new approach, also for web applications, which uses standards-based protocols to enable one application to assert the identity of a user to another, thereby avoiding the need for redundant authentication. Standards to support federation include SAML and WS-Security.
- OpenID

  This is a distributed and decentralized SSO process, where identity is tied to an easily-processed URL which can be verified by any server using the protocol.

software component

The LN software exists of the following separate software components:

- *Message*
- *Report*
- *Label*
- *Function*
- *Business Object*
- *Chart*
- *Integrated session*
- *Additional file*
- *Question*
- *Session*
- *Domain*
- *Table*
- *Menu*
- *Form*
- *Program script*
- *Library*

Software Configuration Management

With software configuration management, a developer can modify and test an own revision of a software component. Using a check-out and check-in functionality, a software component is locked for others developers. This method guarantees that no more than one developer can modify the same software component at the same time.

software language

The language in which the software components, such as forms and reports, are displayed. For each user, the default software language is defined in the LN user data.

solution

In PMC, the smallest, indivisible type of update. A solution is identified both at the distributor and recipient side by a unique solution code. The term individual solution is also frequently used and has the same meaning.

Note: In the PMC software the term solution is often used as an alternative for the term update. A solution can then be an individual solution, which is the smallest, indivisible type of an update, or a patch.

solution status distributor

The following statuses describe the progress of the maintenance of solutions, *Collections*, and *patches*. To keep the descriptions readable, only solutions are described. However, in each case, collections and patches are intended as well, unless explicitly excepted.

The status of the solution is only used at the distributor side. A history of status changes is recorded in the **Maintenance History** table.

To distinguish maintained solutions at the distributor side from imported solutions at the recipient side, the status is cleared when the solution is imported. This also applies to patches, but not to collections. The reason is that a collection is not known as such at the recipient side. Only the solutions contained in the collection are known at the recipient side.

- In progress

  The initial status of the solution upon creation.
- Dependencies defined

  The dependencies between the various solutions are defined, and the solution is ready to be exported. You must select this status manually, because you might want to create dependencies that cannot be created automatically.
- Exported

  The solution is exported. The system handles this status. You cannot set the status manually. However, you can set a solution with the status Exported back to In Progress, Solved, Completed, or Dependencies Defined. If you change anything in an exported solution, the status is always set back to Dependencies Defined, and you must export the solution again. Before you can release a solution, the status must be Exported.
- Released

  The solution is released, which means that the solution is frozen. You can no longer change anything in the solution. To release a solution, you must change the status from Exported to Released, and you must save the solution with this status.

solution status recipient

The following statuses indicate the progress of the installation or uninstallation of a solution or *Patch*.

These statuses are only used at the recipient side, and must not be confused with the *Solution status distributor* at the distributor side.

- Available

  The solution or patch is scanned and available on the system.
- To Install

  The solution or patch is checked and is ready to be installed.
- Saving

  A backup of the components is being saved.

  This status is not applicable for patches.
- Installing

  During the installing process the solution or patch has this status.
- Installed

  The solution or patch is installed.
- To Uninstall

  The solution or patch is checked to be uninstalled.
- Uninstalling

  During the uninstalling process, the solution or patch has this status.

start-up menu

The first menu displayed after LN is started.

The start-up menu code consists of the:

- package code
- module code
- menu code

status bar

A line of information related to the application in the window. Usually located at the bottom of a window.

string

A data structure that contains a number of characters that represent readable text.

strip 8th bit

If you strip the 8th bit from a number, it means to remove the 8th digit in binary notation.

Example

```
1111 1110 (binary) = FE (hexadecimal) =
254 (decimal) -> 111 1110 (binary) = 7E
(hexadecimal) = 126 (decimal)
```

When converting characters between different systems (for example, NLS to TSS), the 8th bit is stripped to obtain the intrinsic character value, which is sometimes needed to address the font.

strip menu

To set a number of menu choices that do not meet certain conditions, to Inactive.

substitution table

A table that consists of two columns: a column with source descriptions and a column with target descriptions. You use substitution tables to do global search and replace actions in LN. The descriptions and helptexts for the software components in LN are searched for the substitution table's source descriptions. The source descriptions that are found, are replaced by the target descriptions.

superseded solution

A superseded solution is a solution for which can be said that all software components are also contained in another so-called *Superseding solution*.

superseding solution

A superseding solution is a solution that contains at least the same software components as contained in one or more *Superseded solutions* and can contain additional software components that are not part of any superseded solution.

A solution supersedes another solution if the following conditions are met:

- The superseding solution contains at least all the components of the other solution.
- The superseding solution contains newer versions of these components.
- The superseded solution is not yet installed. If the superseded solution is already installed, speaking of a superseding solution is illogical.

system administrator

The person responsible for administering the use of a multi-user computer system, communications system, or both. A system administrator assigns user accounts and passwords, establishes security access levels, and allocates storage space.

table

A data structure that is used to store data that consists of a list of records, each entry being identified by a unique key and containing a set of related values. A table contains a number of table fields that belong to a specific domain.

A table code consists of a package code, module code, and three digits.

Example

Table: tc mcs010 **Countries**

This table shows the table fields of tc mcs010 **Countries**:

Code	Label	Length	Data Type
ccty	Country	3	String
dsca	Description	30	Multibyte String
meec	EU Member State	5	Enumerated
...			

table linking

A method of table sharing that stores a table physically in one company (the physical company) and lets other companies (the logical company) use that same table as if the table is their own.

In the following diagram, company 200 shares some of the tables of company 100.

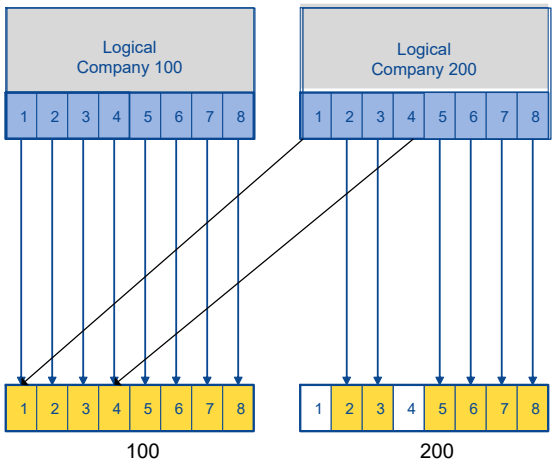


table reference

A link to a table from which something must be selected or where additional information about something is stored.

Example

The **Items - General (tcibd001)** table has a **Country of Origin (tcibd001.ctyo)** field . This field has a reference to the **Countries (tcmcs010)** table.

If you enter a country code in this field, LN does the following:

- Checks whether that code is present in the **Countries (tcmcs010)** table.
- Retrieves the country's description from the **Countries (tcmcs010)** table.
- Displays the country's description on the form.

Hard references are defined in the table definition.

Soft references are not defined in the table definition, but programmed in a program script or dll.

table sharing

A method to use the same data by more than one company. If a user in one company creates, modifies, or deletes a record in a shared table, the effect will be visible in all companies that share the table.

Two techniques can be used to achieve table sharing:

- Table linking.
- Data replication.

template

In the authorization management system (AMS), a method to maintain common data for a group of users with the same role.

text group

A means to define how text must be presented in a window by defining the text editor, default window, and dimensions of the window.

textgroup authorization template

A set of sessions used to print, display, and maintain authorizations that a group of users share for a specific textgroup.

The text group authorizations template is dumped to the user file of all users who use the specific template.

transaction

A logical unit of work, resulting in one or more changes on a database that are carried out as an atomic entity.

transaction notification

A message that states that the data in the source database was changed.

Triton Super Set

The collection of character sets used internally by LN.

TSS character sets include ASCII (ISO 8859-1) characters and a range of multibyte character sets including sets of Japanese and Hebrew characters.

Triton the name of a previous version of the LN software.

tuple

In the business object repository, a tuple is an instance of a component. In the context of databases, a tuple is row in a table. The tuple contains logically related data. The term originated as an abstraction of the sequence: single, double, triple, quadruple, quintuple, ... n-tuple.

undo check-out

A command of Software Configuration Management that deletes the checked-out software component from the Development VRC.

Universal Time Coordinated

A time/date format. LN stores dates and times in UTC format. It stores both date and time in a single Long integer referred to as the UTC long format. This integer represents the number of seconds since 0:00 hour, January 1, 1970 (in UTC).

update

In PMC, an update is a set of changed software components, including PMC metadata, which is required to install the update in a safe and correct way. An update can contain corrective changes or functional enhancements.

Updates can be delivered in four different configurations:

- Solutions
- Collections
- Patches
- Service Packs

update VRC

A physical VRC at the PMC recipient side in which updates are installed. Every update VRC has a *base VRC* linked.

upgrade program

A *DLL* that contains the logic to upgrade a particular table after installation of a more recent *Feature Pack* of LN.

upgrade task

The execution of an *upgrade program* for a particular *company*.

user data template

A template that contains system data and authorizations related to a group of users. The data contained by the template is dumped to the user file of all users who use the specific template.

version

An upgrade of a package, for example, B61. Each change in the functionality of a package results in a new version. A version can contain several releases.

Version - Release - Customer

The version - release - customer (VRC) code is an identification of a stage in the development of the LN software. An example of a VRC is B61_a_ams.

A VRC code consists of:

- Version

  A stage in the development in which a major part of the software is modified.
- Release

  A stage in the development in which a minor part of the software is modified.
- Customer

  An *Extension*, *Localization*, or *Customization* of the software for a single customer or a small group of customers.

A VRC can be derived from a preceding VRC. Every software component that is contained in the preceding VRC and not explicitly modified or set to expired in the current VRC will also be available in the current VRC.

VRC combination

A VRC Combination is defined at the PMC recipient side and consists of a set of related *Update VRCs*. The VRC Combination will guarantee that, if required, the related update VRCs are updated simultaneously at the PMC recipient side. In this way, you can avoid compatibility problems between related update VRCs. For example, if you use the standard application and you also have an extension installed, you will have two update VRCs defined. If a co-requisite dependency is defined between a standard solution and a solution for the extension, PMC installs both solutions in one run if you linked the update VRCs for both standard and extension in one VRC combination.

What's this?

The pop-up Help that tells you briefly the functions of a tool bar button, command button, or menu command.

zoom session

The session in which you can browse through the available records and select a record. A zoom session is an overview session in read-only mode.

You can use a zoom session to enter the code of an existing record (for example, an item, order type, or warehouse) in another session. Click the browse arrow button behind the field or press CTRL+B to start a zoom session.

Index

J

L

M

O

P

Q

R

S

T

U

V

W