Infor Enterprise Server Cloud Edition Administration Guide

Important Notices

Trademark Acknowledgements

Publication Information

Document code	refesagce (Urefesagce)
Release	10.6 (10.6)
Publication date	August 5, 2024

Table of Contents

About this document

LN is a business software solution that consists of applications, tools, and an Enterprise Modeler, all of which work together as a fully integrated system to support all aspects of a business. This document is an Administrator's Guide that describes how administrators must use Infor Enterprise Server to manage and configure LN.

Document summary

LN is a business software solution that consists of applications, tools, and an Enterprise Modeler. All work together as a fully integrated system to support all aspects of a business. This document is an Administration Guide that describes how administrators must use Infor Enterprise Server to manage and configure LN.

"(CE)" and "(OP-CE)" suffixes in section titles

Each section title in this guide has one of these suffixes:

- "(CE)": Indicates that the section is only applicable in a cloud scenario.
- "(OP-CE)": Indicates that the section is applicable in a cloud scenario and in an on-premises scenario. The topic is also present in the on-premises edition of the *Infor Enterprise Server Administration Guide*.

Comments?

We continually review and improve our documentation. Any remarks/requests for information concerning this document or topic are appreciated. Please e-mail your comments to documentation@infor.com .

In your e-mail, refer to the document number and title. More specific information will enable us to process feedback efficiently.

Contacting Infor

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

If we update this document after the product release, we will post the new version on the Infor Support Portal. To access documentation, 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
To Get Started

1

Introduction (OP-CE)

This Administrator's Guide describes how you must use Infor Enterprise Server to manage and configure LN, a business software solution that includes applications, tools, and an Enterprise Modeler. All of these components work together as a fully integrated system and supports all aspects of a business.

This preface describes:

- Who must read this document
- Administration in perspective
- How to use this document
- The setup of this document
- Additional information

Who must read this document (CE)

This document is intended for the system administrators who set up, configure, and manage the LN software. The Administrator's Guide describes how to use Infor Enterprise Server to set up an LN environment with LN users, user authorizations, devices, and so on.

You can use this document as a Reference Guide. The "System administrator tasks" chapter, describes the responsibilities of system administrators.

Administration in perspective (CE)

This document describes the main functions and objectives for the administration of LN. This document provides information about system information in LN and how system administration is related to the other parts of LN. You must perform several administrative tasks. For example, you must define 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.

How to use this document (OP-CE)

The following section, “Setup of the guide,” outlines the content of each chapter in this document, which you can use as an overview of LN ’s administration.

The structure of the chapters is typically as follows:

1. Overview section: Describes the administration concepts and any necessary additional information.
2. A procedures section: Describes the aim, the prerequisites and the result of the procedures, and contains a summary of the procedure steps.

If you are familiar with the concepts of a chapter, you can skip the overview section and proceed directly to the procedures section.

Note

- For detailed information on the procedure steps, refer to the "Procedure Details" sections in the **Infor Web Help**.
- For detailed information on the sessions that are used in the procedures, refer to the session help.

Setup of the guide (CE)

These topics are described:

The system administrator’s tasks

The tasks and responsibilities of the system administrator are, for example to assign user accounts and passwords, establish security access levels, allocate storage space, and watch for unauthorized access.

User management

The LN administrator uses LN ’s User Management to enable users to work with LN and to give these users authorizations related to their role.

These procedures are described:

- How to create a user account

- The Authorization Management System (AMS)

Audit Configuration Management

An overview is provided of the steps you must take to configure the audit settings. According to this configuration the audit trails are created. The audit configuration is based on audit profiles. In an audit profile, you define which tables and fields are audited, and when the table and fields are audited.

The audit trail is stored in sequence files, which are generated for each combination of company and table.

Also, the procedures for various other Audit Management features are described, such as how to import and export profiles, and generate reports.

Note

Several important parameter tables must be audited mandatory. The audit history data of those tables will be used by Infor Support.

Device management

One of LN ’s most important tasks is to generate output. Raw data is fed into LN and processed into reports, sales invoices, salary check, and so on.

This chapter describes how to set up printers for LN and the related sessions.

Text management

You can use text for various purposes in LN. You can, for example, use text to provide information about LN itself. For example, technical documentation and release notes for software components. You can also use text to provide information on the data stored in the tables of the database: For example, a text that is linked to a certain record can contain information about an item or sales order pertains to that record.

Described are these procedures in regard to text management:

- The text parameters procedure, which defines the basic parameters for the use of text
- The text maintenance procedure

Job management

You can use LN ’s job management to schedule jobs at non-peak hours to improve the overall system performance in a heavily loaded environment. You can schedule the jobs to start processes periodically, at a defined interval, or immediately. Use job management for print and processing sessions.

Additional information (CE)

The Tools section in the LN help can be helpful to administrators. This section contains online manuals on various important topics, for example:

- User Management
- Audit Management
- Device Management
- Text Management
- Job Management

Chapter 2
System Administrator Tasks

2

System administrator tasks introduction (CE)

This section describes the system administrator's tasks and responsibilities, such as to assign user accounts and passwords.

To explain these tasks and responsibilities fully, this section includes:

- An overview of system management
- The system administrator's tasks

Overview of system management (CE)

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

Administration of your LN installation

Because LN is a large package and can have many users, someone or some group of people must manage the package. These individuals are referred to as the LN system administrators.

The LN system administrators must perform the following tasks on an installed LN system:

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

System management and application management differ by company. Usually, these tasks are full-time jobs.

Depending, for example, on the computer system and the number of users, a system can be managed by a system manager and an assistant system manager. Most large companies, usually have more than one system manager, each with specific responsibilities. In these cases, an application administrator is usually in place to manage the applications. In smaller companies, however, one person performs system management and application administration.

System administrator tasks (CE)

The application administration consists of a number of 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 will be performed by contacting one of the Global Support Centers.

Reorganize tables

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

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.

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.

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 a company number For details on this procedure, refer to To copy a multicompany structure in the “Enterprise Server” section in the **Infor Web Help**.

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 user data reports
- The updates
- Changes in the system
- The error messages and solutions

Chapter 3
User Management

3

User management introduction (CE)

This chapter describes 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.

This chapter includes the following:

Centralized user management through IFS

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

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 (CE)

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

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

This overview includes the following topics:

- LN user types (OP)
- *LN user authorizations (OP-CE) (p. 22)*
- Role-dependent authorizations (OP)
- *Non-role-dependent authorizations (CE) (p. 31)*

LN user types (CE)

LN users are divided into the following two groups:

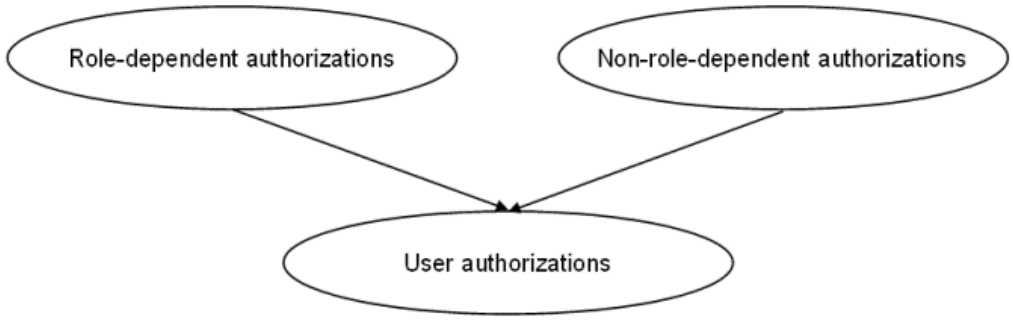
- Normal users: These users have restricted authorizations to start sessions, access table fields, work with data that pertains to specific company numbers, and so on.
- Super users: These users have unrestricted authorizations. A system administrator is a good example of a super user.

LN user authorizations (OP-CE)

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:



Schematic overview of the authorization concept in LN

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 (DEM, AMS and Enhanced AMS)

Use these methods for defining employee roles:

- Dynamic Enterprise Modeler (DEM)
- Authorization Management System (AMS)
- Enhanced AMS, that fully integrates DEM with AMS

DEM

In DEM, an organization can be modeled, including DEM Business processes. DEM 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 DEM employee. This employee has one or more DEM Roles. These DEM Roles are linked to one or more DEM 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 DEM menu, the Process browser. When an employee runs LN sessions from the Process browser, the modeled DEM authorizations are used to authorize all sessions in the menu. The DEM 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 DEM 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 decide what is allowed to execute in the LN Menu browser, and what the level of authorization is.

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

Enhanced AMS

This method is available to link DEM 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 DEM modeler, we recommend that you use the Enhanced AMS with DEM.

To enable Enhanced AMS with DEM:

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

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

DEM session occurrence 1	DEM session occurrence 2	After aggregate
No authorization	Display	Display
Full	Display	Full

If you use Enhanced AMS, DEM subapplications are handled differently than if you use only DEM:

- If you only use DEM, a subapplication that is not specifically modeled (that means, not given an authorization level in the DEM module) has the same authorization level as the main session it is part of. That is, the authorization level is inherited.
- 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 DEM and AMS is used. This table shows an example:

DEM	AMS	Runtime
Display	Full	Display
Full	Display	Display
NA or blank	Full	NA
Full	NA or blank	NA

Suppose, a subapplication is not specifically modeled in DEM, 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 DEM 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 DEM 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 Act (officially titled the Public Company Accounting Reform and Investor Protection Act of 2002).

These sessions are available to print the session authorizations:

- **DEM**
  Print DEM 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.

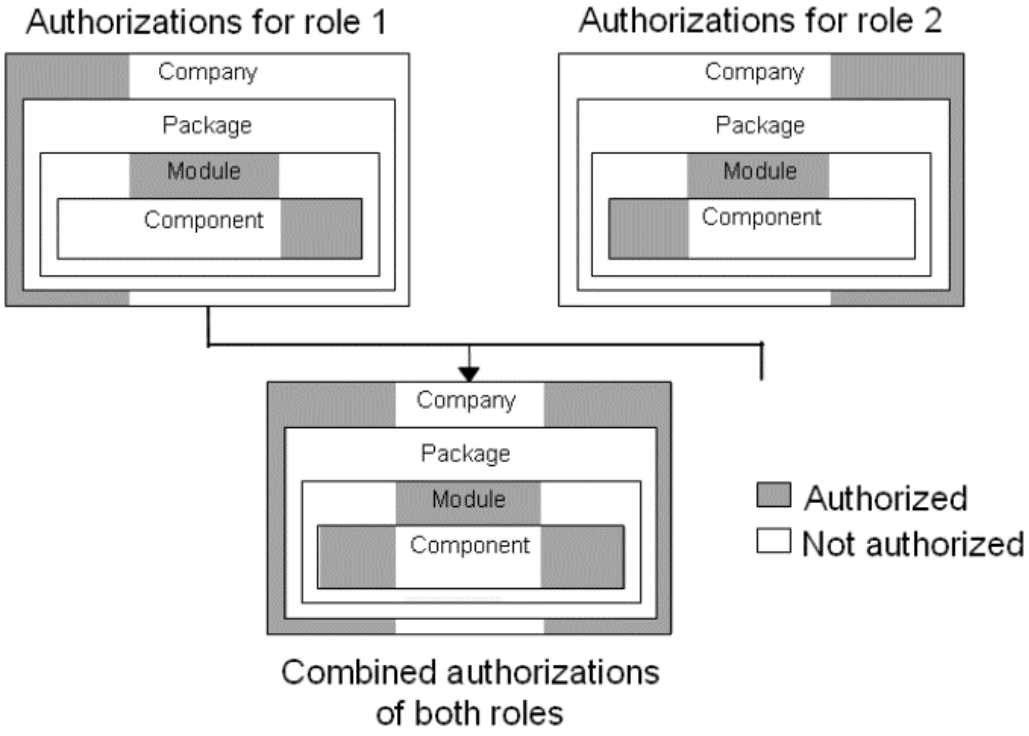
Role-dependent authorizations (CE)

From a user point of view, 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 quickly 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:



Example of combined authorizations for more than one role

- **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 will ultimately have permission to carry out the database actions.

Role-dependent authorization types in LN

You can define the role-dependent authorizations at the following component levels in LN:

- 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 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 priorities in the following table show that the session authorization with the highest priority (1) is stated at the most specific level and 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:

	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

You can define the session authorizations with these sessions in LN ’s AMS:

- 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 authorization priorities in the table show that the table authorization with the highest priority (1) is stated at the most specific level and 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.

	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 and 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.

Note

For details on the Business Object Layer (BOL), refer to To Model a Business Object in the Infor Enterprise Server **Web Help**

You can specify the library authorizations at several levels. For example, 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 priorities in the following table show that the library authorization with the highest priority (1) is stated at the most specific level, and the lowest priority (3) is stated at the most global level:

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 (CE)

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 are not dependent on the employee's role. The templates can also contain additional parameters that are required for developers to be able to customize software components in LN.

The templates contain data a group of users share. However, the templates are linked to individual users. The templates in LN offer a user-friendly method to add new data quickly 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 the following 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 the following 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 use the template parameters that application developers use. For example, you can specify the compiler options that can be used when you compile a program.

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:

- 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.

If you select the **Authorization for all Modules** and **Authorization for all Languages** check boxes in this template, the users who are linked to the template are authorized to maintain and develop software components in all modules in LN 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 necessary 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. If you select the **All Companies** check box in these sessions, 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, LN uses the default text group defined in this template. 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

You can use this template to define the following authorizations for each text group: the use, update, and read. 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, as well as 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.

User-related procedures (CE)

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

With these procedures, you can:

- Create LN users.

- Maintain the user’s developer authorizations.

Procedure result and prerequisites

Result

As a result of these procedures, the user can start LN, use the Menu Browser, and start sessions. Developers will have the proper 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 roles and templates to which you can link the normal users are already defined. If these components are not defined, refer to the *To create roles and authorizations (OP-CE) (p. 35)* and *To create templates (OP-CE) (p. 36)* 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.

To create LN users (CE)

To be able to work with LN, a user must have a user logon, a password, and the proper authorizations. The authorizations are dependent 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.

Procedure summary

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)

To change the password for developer authorizations (OP-CE)

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

Procedure summary

This section shows the procedure step and the corresponding session:

Change the password for developer authorizations - Change password for Role and Developer authorization (ttadv0143m000)

To create roles and authorizations (OP-CE)

Initially, normal users do not have any authorizations for LN. Therefore, you must define the role-dependent authorizations for their function. This section provides information and instructions on how you can use LN ’s Authorization Management System (AMS) to create roles for normal users and the associated role-dependent authorizations.

You can use the authorization management system to:

- Create roles and authorizations.
- Create templates.

Procedure result and prerequisites

Result

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

Prerequisites

The following 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 following 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. Employees of Global Enterprises must be able to use the company’s data dictionary and the databases of both offices.

For detailed instructions on how to enter data in the described sessions, refer to the online help of the sessions

Procedure introduction

The authorization management procedure is split up into several smaller procedures, which the system administrator can use 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 easily maintainable, the system administrator defines one standard role with the most basic authorization, as well as additional roles for specific tasks.

The authorizations 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 makes sure 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)
   Via 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)

   For details on the authorization types, see Role-dependent authorizations (OP).
3. Convert the user file to the runtime data dictionary (ttams2200m000)
4. The relevant users must log off and log on again.

To create templates (OP-CE)

A template contains common data for a group of users. This section provides information that you need to create templates. In these templates, you can define the non-role-dependent authorizations for a group of normal users. You can also enter additional template parameters that a group of users share.

Procedure result and prerequisites

Result

By using templates, the system administrator has 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)

This section uses the User Data template 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.

As soon as 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.

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

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)
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 (CE)

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 session is only supported when 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 will stop the selected bshell. Note that when you are logged on as a **Normal User** you only can only stop your own processes. To be able to terminate the processes of other users you must log on as a **Super User**.

Chapter 4
Audit Configuration Management

4

Audit configuration management introduction (CE)

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.

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.

This section provides the following:

- *Audit trail (CE) (p. 40)*: These settings provide the information that is required to create the audit trails.
- *Audit configuration procedure (OP-CE) (p. 40)*: This section provides an overview of the steps you must take to configure the audit settings. According to this configuration, the audit trails are created.
- Audit - Additional functions (OP): This section contains some information about other functions, such as importing and exporting profiles, and generating reports.
- *Audit - General remarks (CE) (p. 41)*: This section lists a number of important facts and rules that you must bear in mind.
- *Audit - Specific issues (CE) (p. 42)*: These sections describes what happens if you convert new audit settings to run time while not all users have left their bshell.

Audit trail (CE)

Before you can create an audit trail, the following 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. Because the maximum number of sequence files for each table/company combination is 999, and you cannot delete the currently active file, you must choose a file size that enables you to delete old files, while keeping 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 security settings for reading, maintaining, and deleting the sequence files. Define the security settings in the Audit Trail Security (ttaud3137m000) file.

Note

To activate these settings, you must use the Create Runtime Audit Definitions (ttaud3200s000) session, with the correct check boxes selected, 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 (OP-CE)

Introduction

This section provides an overview of the steps you must take to configure the audit settings, assuming that no audit settings are yet present.

Procedure summary

The following list shows the procedure steps and the corresponding sessions.

1. Define audit categories - Audit Categories (ttaud3100m000)
2. Define company groups - Company Groups (ttaud3140m000)
3. Define which companies are related to the company groups - Companies by Company Group (ttaud3145m000)
4. Define 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 (CE)

The audit configuration management sessions provide the following additional functions:

You can export and import profiles through the following 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, by means of the following sessions:

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

To maintain the generated sequence files, you can use the following 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 (CE)

The commands that cause a table transaction to be audited, are only the commands that affect the table data, that is, 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, which means that you can only define the tables and fields that must be audited, but not 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 sessions concerned.

Because you can use tables and fields in various profiles with conflicting settings, the following rules determine which setting take precedence over the conflicting setting:

- A higher level takes precedence over a lower level. Therefore, if you define in profile A that all tables in a module must be audited, but in profile B that only one table in this module must be audited, the result, if you convert these profiles to run time, is that all modules in the package concerned are audited.

- The setting **Always** takes precedence over **Changed**. Therefore, if profile A defines that a field must only be audited, when the profile is changed, and profile B defines that this field must always be audited, the field is always audited.

Audit - Specific issues (CE)

The following sections describe the impact of several changes you can make in the audit settings. The impact of a change depends on the specific situation. The situations described arise if changes in the audit settings are converted to run time while not all users did leave LN. As a result, some users create audit trails based on the old configuration, while users that start afterwards create audit trails based on the new configuration.

Changes in the profiles

The following table summarizes the impact of the changes for a particular table in a company:

Impact of 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.
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, while 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. As a result, users with the old settings can be forced to restart LN.

Notes

- If you switch field-specific auditing for a table on or off, and this results in a different number of fields to be audited, the impact 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 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 will grow until this new size. If the file size was diminished, and the current file already exceeds this size, a new file will be created the next time a transaction is logged.

Chapter 5
Device Management

5

Device management introduction (CE)

LN reports can be printed on various types of devices, such as output files and printers. This chapter describes how to set up printers for LN and how to manage the printer environment.

This chapter describes the following:

- *Device management overview (CE) (p. 45)*
- *Device management setup procedure (CE) (p. 47)*
- *Device management maintenance procedure (CE) (p. 48)*

Device management overview (CE)

You can use the device management functionality to set up print devices for LN and to print LN reports to those print devices.

Raw data is fed into LN and processed into reports, sales invoices, paychecks, and so on.

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

LN 's device management supports an entire range of device types.

In LN, an output device can be one of the following:

- **screen**
  The output data appears 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.
- **windows server printer**
  This device type provides the possibility to print through the Cloud Printing Service. The Cloud Printing Service can print to a printer or to a fileshare.
- **External Reporting Services**
  A device to print LN reports by means of a report design that is stored on an external Report Server. This Report Server will render the report. The destination of the report depends on the arguments filled in the **Argument** field in the Device Data (ttaad3100s000) session.

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 will be converted into a special format.

The following fields must be used to specify the conversion parameters:

- **Device**
  You can choose a unique name for a device.
- **4GL program**
  Depending on which print output format you want to use, you must specify the name of the conversion script, such as ttstpconv.
- **Argument**
  You must use this field to specify the print output format, such as ASCII or XML.

For details on how to configure a device refer to 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.

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, if you want to print paychecks, you must define the characteristics of the preprinted checks that must be loaded in the printer.

Device queue data

Every print request is added to the device queue. LN uses the status of the print job in the device queue to do the following:

- 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.

Device management setup procedure (CE)

LN ’s Device Management contains the procedures that you can use to create devices and specify the paper types and fonts for the printouts.

Additional procedures are included in device management, which you can use to:

- Define paper types
- Define the fonts for the paper types
- Create devices

After the procedure is completed you can use a range of devices to view or print data that LN processes.

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:

Step 1: Define paper types

You can define specific paper types for every type of output. For example, if you want to print paychecks, you must define the characteristics of the preprinted checks that must be loaded in the printer.

Session: Paper Types (ttaad3110m000)

Step 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.

Session: Fonts by Paper Type (ttaad3111m000)

Step 3: Create the devices

You must create the devices to which LN can send processed data.

Sessions: Device Data (ttaad3500m000) and Device Data (ttaad3100s000)

Device management maintenance procedure (CE)

Procedure aim

After you complete the Device Management setup procedure, you can print LN ’s processed data. LN ’s device management also gives you the tools to maintain the device management data.

You can, for example:

- Maintain the device queue.
- Cancel print requests from users and devices.
- Change the device status.
- Purge the device queue.
- Change paper types.

This section describes only the most important steps in the device management procedures. For detailed instructions on how to enter data in the sessions described in the procedure steps, refer to the online session Help.

To maintain the device queue

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. However, 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.

As a result , the print request receives the Canceled status.

Sessions: Cancel Device Requests by User (ttaad3220m000) and Cancel Device Requests by Device (ttaad3230m000)

Windows printer devices (CE)

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.

Microsoft Excel devices (CE)

You can define devices that allow you to print reports to an MS Excel spreadsheet.

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 **4GL Program** empty.
   d. In the **Argument** 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 filename has this format: `[report name]_[company number]_[yyyymmdd-hhmmss].xlsx`

This is a sample filename: `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 filename has this format: `[report name]_[company number].xlsx`

  This is a sample filename: `ttaad340001000_000.xlsx`

e. 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.

Document Output Management (OP-CE)

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 6
Text Management

6

Text management introduction (OP-CE)

The information about LN and its functionality is written with the tools supplied by LN ’s Text Management. This chapter describes the procedures that are required to create and maintain an environment that you can use to create texts in LN.

This chapter describes the following:

- Text management overview
- Text management parameters procedure
- Text management maintenance procedure

Text management overview (CE)

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 provide information about the data stored in the database tables.

You can, for example, do the following:

- 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.

Description of text-related templates

A normal user must have some basic text authorizations. You can define the following 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 necessary data and authorizations are defined in the text parameters. These text parameters are defined in the following 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, LN uses the default text group 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.

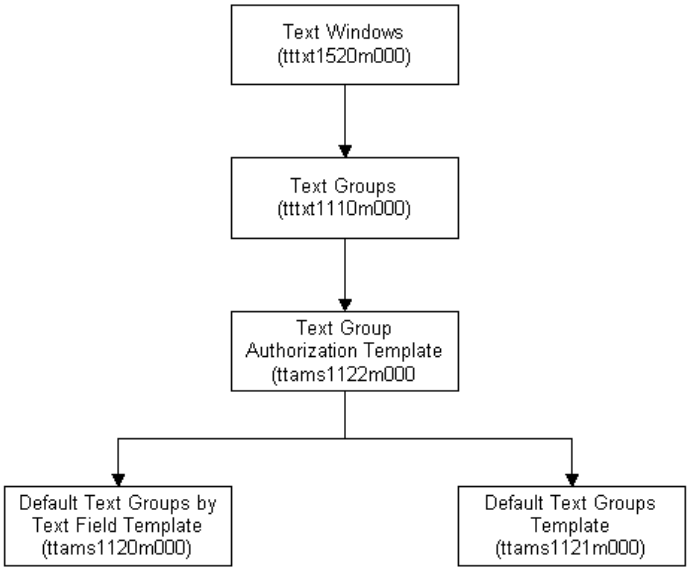
Text management parameters procedure (CE)

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.

This diagram shows the steps to define the text parameters:



Step 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. However, you must make a selection in all fields because you can also use other editors for the text group.

Sessions: Text Windows (tttxt1520m000) and Text Windows (tttxt1120s000)

Step 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.

Session: Text Groups (tttxt1110m000)

Step 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.

Session: Text Group Authorization Template (ttams1122m000)

Step 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:

Session: Default Text Groups by Text Field Template (ttams1120m000)

Step 5: Create the template for the default text groups

In the default text groups template, you must specify the text group LN must use if you have not specified a specific text group for a text field.

Session: Default Text Groups Template (ttams1121m000)

Text management maintenance procedure (CE)

Procedure aim

This section describes the following procedures:

- The procedure to create a range of text numbers
- The procedure to remove unused texts

This section describes only the most important steps in the text management procedures. For detailed instructions on how to enter data in the sessions described in the procedure steps, refer to 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. Make sure that, for each site, you define unique ranges for the text groups to avoid conflicts with identical text numbers on separate sites.

Session: Text Number Range by Text Group (tttxt0130m000)

To remove unused texts

Unused texts are texts that are no longer linked to a record.

A text becomes unused, for example, if you remove the record, for example, the sales order or the item, to which the text was linked, or if 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.

Session: Delete or Print Unused Texts (tttxt1210m000)

Chapter 7
Job Management

7

Job management introduction (CE)

You can use job management in LN to schedule jobs 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, that run without user interaction. The sessions 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 (CE)

To create a job, you must specify basic job data and link sessions to the job. In the basic job data you specify whether the job is periodical. For periodical jobs, you specify how the job will be scheduled.

You can use a job to run one or more sessions as follows:

- 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 the following two job types:

- **Periodical jobs that are carried out multiple times**
  For a periodical job, you must specify a time interval or a calendar is specified. 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 (CE) (p. 59)*.

Shared job data tables (OP-CE)

Typically, each company stores its own basic job data. As a result, a job runs for a particular company. However, 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 the following 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 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 in the user interface to define the job for each company. You can then define the job data for the other companies.

Job execution (CE)

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**.

Free status

If the job's status is Free, you can start the job through the Activate Job (ttaad5203m000) session.

Activate Job (ttaad5203m000)

If you start the job before the time and date of execution, the job's status changes to **Waiting**. While 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 on the Menu browser's **Option** menu to switch to the required company.

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.
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 will 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, 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 following sessions in the job will not be 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 (OP-CE)

To handle dependent jobs, you can create a job group.

Job groups are convenient to use when running job 'A' depends on the result of job 'B' and it is difficult to plan at what time job 'A' must be scheduled.

This session is a multi-main table session. It shows multiple tables and allows you to 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 start the Job Groups session directly.

Creating a job group:

1. Start the Job Groups (ttaad5140m000) session for creating 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 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 of 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 (OP-CE)

When the execution of a job stops, for example, when the job completes successfully or when a runtime error occurs, information is written to a history log. The job history contains information, such as the date and time of the execution and the reasons why the job and its associated session 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.

Job data procedure (CE)

LN ’s Job management contains the procedures that you can use to create jobs, and define the time and date in which the job must be carried out in the job calendar.

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 a job calendar and the basic job data. In LN, you can start periodical jobs according to a schedule that is defined in the job calendar or a regular interval. You can define the interval in the basic job data. After you define the basic job data, you can add sessions to the job. The default settings of the sessions in the job can be maintained, when required, with an optional step in the procedure.

Step 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.

Session Calendars (ttaad5106m000).

Step 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.

Session Dates by Calendar (ttaad5107m000)

Step 3: Define the basic job data

Before you can add sessions to a job with the Add Session to Job (ttaad5102s000) session, you must first define the basic parameters for the job. To make a periodical job non-interactive, in the sessions Job Data (ttaad5500m000) and Job Data (ttaad5100s000), you can specify a predefined job calendar or a regular interval to start the job.

If you select the **Suspend Job until Next Execution** check box, the job is suspended until the next execution date. 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. For jobs started by the Activate Job (ttaad5203m000) session, the job’s status is set to **Free**. To restart the job, you must manually start the Activate Job (ttaad5203m000) session.

Step 4: Add sessions to a job

A job must contain one or more sessions, which must be carried out according to the data specified in the session Job Data (ttaad5100s000) session. Typically, a job starts print or process sessions.

Under Error Handling, you can enter 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, a number of sessions of the job if an error occurs during the execution of the session.

Step 5: 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.

Session Sessions by Job (ttaad5501m000).

Job handling procedure (CE)

Procedure aim

With the job handling procedures, you can start, block, cancel, queue, and release jobs.

Procedure result and prerequisites

Result

This procedure results in a job that runs according to the job calendar or interval that is defined in the job’s basic data, without user interaction. This helps you to use LN ’s resources more efficiently.

Prerequisites

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 starts the job. If the job’s status is Free, you can start the job by running the Activate Job (ttaad5203m000) 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 prior to 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.

If you must start a job that belongs to more than one company, you must switch to the required company with the **Change Company** command, before you start the job.

Chapter 8
Multilanguage Application Data

8

Multilanguage application data (CE)

You can run multiple languages of the LN software, such as Chinese, English, and French in one LN environment. Also, 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.

Language types

This table shows the language types:

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 the following 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.

ISO standards

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 (_).

For example:

Data Language	Description
de	German
en_GB	English_UNITED KING-DOM
en_US	English_UNITED STATES
he	Hebrew
it	Italian
ja	Japanese
nl	Dutch; Flemish
pt_BR	Portuguese_BRAZIL
pt_PT	Portuguese_PORTUGAL
zh_CN	Chinese_CHINA (Simplified)
zh_TW	Chinese_TAIWAN (Traditional)

Configuration Multilanguage Fields Support (CE)

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.

Step 1: Enable support for Multilanguage Fields

To enable support for Multilanguage fields:

1. Start the Parameters Multi Language Support (ttadv4106m000) session.
2. Select the **Multi Language Fields Support** check box.
3. Save the changes and close the session.

Step 2: Define data languages

To define data languages:

1. Start the Data Languages (ttaad1111m000) session.
2. 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.
3. Define the base language: Select a data language and, on the appropriate menu, click **Make Base Language**.
   Note: The base language is used as a fall back language. Application data is displayed in the base language if both of the following 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.
4. 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.

   Important:
   - 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.
5. Close the session.

Data Languages have a status that is used to control their life cycle. For details, refer to "Life cycle of a data language" in the online help of the Data Languages (ttaad1111m000) session.

Step 3: Link data languages to software languages

You must link data languages to the corresponding software languages.

To link a data language to a software language:

1. Start the Software Languages (ttaad1510m000) session.
2. To start the Software Languages (ttaad1110s000) session, double-click a software language. In the **Data Language** field, enter 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 you leave the **Data Language** field blank, LN displays the application data in the data language defined in the user data of an LN user, or in the base language.
3. 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.

Step 4: Register Multilanguage Tables and Table Fields

Register the tables and table fields that must be multilanguage enabled.

1. Start the Registered Tables with Multi Language Fields (ttadv4137m000) session.
2. Add the tables you want to be Multilanguage Enabled.
3. Double-click a table. The Registered Multi Language Fields (ttadv4138m000) session starts.
4. For each field you want to be multilanguage enabled, select the **Enabled** check box.
5. Save the changes and close the sessions.

Step 5: Convert to runtime

To convert the multilanguage settings to the runtime data dictionary:

1. Start the Manage Operational Status (ttmtm3501m000) session.
2. Click **Admin Only** to switch the environment to Admin only mode.

   In this mode, normal users can no longer log on to the environment. Users that were already logged on are forcedly logged off.
3. Start the Convert to Runtime Data Dictionary (ttadv5215m000) session.
4. Select the **Tables** check box and the **Reconfigure Tables** check box.
5. For packages and tables, specify the appropriate selection ranges.
6. Click **Convert to Runtime**
7. Optionally, start the User Data (ttaad2500m000) session, and define a data language per user. The data language specified in the user data takes precedence over the data language linked to a software language. For more information, see "Language types" in *Multilanguage application data (CE) (p. 63)*.
8. To actualize the related domain and table definition changes, reinitialize the shared memory:
   a. Log off.
   b. Wait for two minutes, so that the shared memory can be stopped correctly.
   c. Log on again. The shared memory is reinitialized.

      Do not stop the logon process by closing the browser (tab) during the logon process. You must wait until the LN menu is visible.
9. Start the Manage Operational Status (ttmtm3501m000) session.
10. Click **Bring Online** to bring the environment online again, so that normal users can log on again.

Note

If you use the DEM 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 (CE)

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 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.

To ensure that a user can switch to another data language, you must select the **Allow Changing of Data Languages** check box in the user's user data template properties (User Data Template (ttams1110m000) session).

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. When 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 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. When the user inserts a record, the descriptions of the multilanguage enabled table fields are inserted in the other language code, and are automatically copied to all other available data languages and the base language.

2. For each table 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 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.

   You must also check whether the descriptions stored in the original language code, that is, the language code in which the records were created, are still correct.

   To remove the corresponding records in the check report, save a dummy change of these descriptions.

Chapter 9
Sensitivity Labeling

9

Sensitivity labeling (OP-CE)

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 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.

To configure sensitivity labeling:

Step 1: Enable sensitivity labeling

Start the Tools Parameters (ttaad0100m000) session and select the **Sensitivity Labeling enabled** check box.

Step 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.

Step 3: Assign sensitivity labels to table fields

Start the Sensitivity Level of Table fields (ttaad3151m000) session. This session allows you to assign sensitivity labels to tables or table fields.

With the Print Impacted Components (ttaad3450m000) session you can print a list of reports which will get a sensitivity label based on the defined fields.

Step 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.

Step 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 (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 the following font settings:

  - Font size: 1.5 times the default font of the report.
  - Font style: bold.

Step 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** checkbox in Tools Parameters (ttaad0100m000) is changed.

Chapter 10
Customer Defined Fields

10

Customer defined fields (CE)

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**.

Configuration

To configure customer defined fields:

Step 1: Activate the CDF feature.

1. Start the Customer Defined Fields Parameters (ttadv4590m000) session.
2. Select **CDF Active** and click **OK**.

Step 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.

Step 3: Convert to runtime.

1. Start the Manage Operational Status (ttmtm3501m000) session.
2. Click **Admin Only** to switch the environment to Admin only mode.

   In this mode, normal users can no longer log on to the environment. Users that were already logged on are forcedly logged off.

3. Start the Convert to Runtime Data Dictionary (ttadv5215m000) session. Convert the customer defined fields and the related domains to the runtime data dictionary.
4. To actualize the related domain and table definition changes, reinitialize the shared memory:
   a. Log off.
   b. Wait for two minutes, so that the shared memory can be stopped correctly.
   c. Log on again. The shared memory is reinitialized.
      Do not stop the logon process by closing the browser (tab) during the logon process. You must wait until the LN menu is visible.
5. Start the Manage Operational Status (ttmtm3501m000) session.
6. Click **Bring Online** to bring the environment online again, so that normal users can log on again.

Note

If you use the DEM Content Pack with Infor 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.

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.
- Customer defined fields cannot store application data in multiple data languages.
- There is no direct limitation on the number of CDFs in a table. However, 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.

Personalization

A session can contain two types of customer defined fields:

1. 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).
2. 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 "Application Personalization".

Related documentation

For more information about customer defined fields, see "Customer Defined Fields" in the *Infor LN Extensions Development Guide*.

Chapter 11
Scrollbars

11

Scrollbars (CE)

LN sessions support these types of scrollbars:

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.
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.

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, you can:

- Disable the new scrollbar for sessions that contain many records.
  For details, refer to To disable the scrollbar for sessions with 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.
  For details, refer to To disable the scrollbar for any session.

If you start a session with a disabled scrollbar, the old scrollbar is displayed.

To disable 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, enter 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.

To disable 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, for which you want to disable the new scrollbar, to the list.

   For details, refer to the session help.
4. Save the changes and close the sessions.

Chapter 12
Document Authorization

12

Document Authorization Overview (CE)

Overview

Database Change Management (DBCM) supports Document Authorization, using ION Workflow. Document Authorization is about approving or rejecting changes 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, and/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. The rest of the ERP system does not know this version. This checked-out version is a kind of scratch version. A user can change anything, but the changes will not become available to the rest of the system until the user submits the changes, and someone approves these changes. It is also possible to 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 *Procedures for checked-out objects (OP-CE) (p. 86)* 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. See Overall procedure.

If the Model describes two Object Types A and B, and you use Document Authorization for Object Type A, DBCM will 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 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. See Application changes.

Application changes

To support Document Authorization, the application must be adapted.

Once 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, so 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 which 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 via 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 will not be 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 (OP-CE)

To import the standard model:

1. Start the Models (ttocm0101m000) session.
2. On the appropriate menu, select **Import Standard Model**.

Creating a model (CE)

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 the following sections.

Upon completion of this procedure, you must deploy the model. See Deploying a model (OP).

Note

When you customize a model, you may also have to 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. You can only use tables from the Extensibility package (tx) as a root table.
4. Save the new object type.

Adding actions to the object types

You can add one or more actions to each object type. You can only add actions for object types with a root table from the Extensibility package (tx).

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 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, you must 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 (CE)

Step 1: Generate a deployment.

1. In the Models (ttocm0101m000) session, select the model.
2. On the appropriate menu, select **Deploy...**. The Deploy Model (ttocm0101m100) session is started.

3. Specify a description for the deployment.
4. Click **Deploy**. The deployment is generated.

Step 2: Deploy the actions of the deployment.

1. Start the Deployments by Package Combination (ttocm0111m000) session.
2. Select the deployment and, on the appropriate menu, select **Deployed Actions**. The Deployed Actions (ttocm0112m000) session is started.
3. 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**.
4. Save the changes and close the Deployed Actions (ttocm0112m000) session.

Step 3: Activate the deployment.

1. Start the Manage Operational Status (ttmtm3501m000) session.
2. Click **Admin Only** to switch the environment to Admin only mode.

   In this mode, normal users can no longer log on to the environment. Users that were already logged on are forcedly logged off.
3. In the Deployments by Package Combination (ttocm0111m000) session, select the deployment.
4. 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; if the session is password-protected, you are prompted to specify the password.
5. To actualize the related domain and table definition changes, reinitialize the shared memory:

   a. Log off.

   b. Wait for two minutes, so that the shared memory can be stopped correctly.

   c. Log on again. The shared memory is reinitialized.

      Do not stop the logon process by closing the browser (tab) during the logon process. You must wait until the LN menu is visible.

   The changes are now fully actualized.
6. Start the Manage Operational Status (ttmtm3501m000) session.
7. Click **Bring Online** to bring the environment online again, so that normal users can log on again.

Procedures for checked-out objects (OP-CE)

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 13
LN Software Maintenance

13

PMC introduction (CE)

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 software updates (Individual Solutions).

The PMC module manages the installation of Individual Solutions.

Without a tool such as PMC, software updates could be installed irrespective of other software updates that were already installed. Technically, an update that contains an earlier version of a software component than the version already present on your system, could be installed. In this case, you lose the updates contained in the original software components.

The PMC module contains two sub-modules.

- PMC distributor
- PMC recipient

PMC Benefits

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

- Automatic checks for updates for any previous dependencies, to ensure that these exist on the system on which you install the update.
- Analysis of potential conflicts with system customizations at installation time, which enables the user to identify any conflicting customizations.
- Option to maintain copies of all previous software components. This feature enables the user to roll back updates easily to any point in history.
- Overview of all 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 overview (CE)

This chapter provides an overview of the Product Maintenance and Control (PMC) module.

This chapter contains the following sections:

- PMC Architecture
- Where to find the PMC module
- Dependencies
- Individual Solutions
- PMC Distributor functionality
- PMC Recipient functionality

PMC Architecture

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.

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, complete these steps:

1. Log on to LN.
2. On the ERP Menu Browser, click **Tools --> Software Installation --> Miscellaneous --> Recipient**.
3. Click **Setup, Operational, and Miscellaneous**.

Alternatively, complete these steps to navigate to the recipient part of the PMC module:

1. Log on to LN.
2. On the ERP Menu Browser, click **Tools --> Application Development --> Product Maintenance and Control --> Recipient**.

3. Click **Setup, Operational and Miscellaneous**.

To navigate to the distributor part of the PMC module, complete these steps:

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

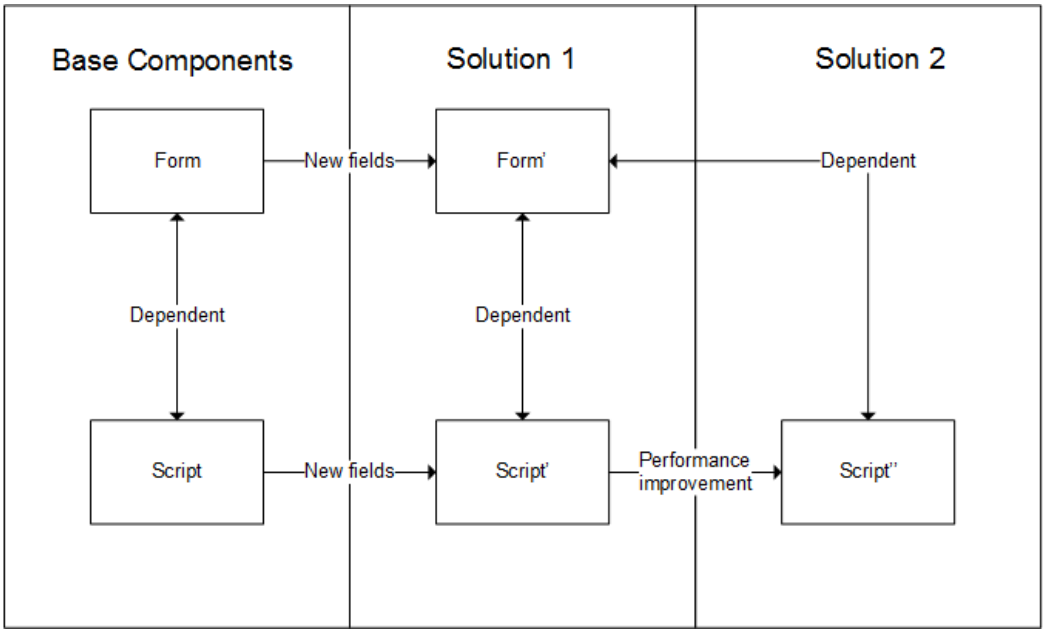
Dependencies

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

Three dependency types exist:

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

This diagram shows a pre-requisite dependency: solution 1 must be installed before solution 2.



The following section describes a customer scenario.

Individual solutions

A PMC Distributor creates and publishes individual solutions.

After you download and scan a solution, the PMC recipient checks the dependencies with other solutions. In some cases, 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.

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.

The following is 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.

PMC Recipient functionality

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

The following is an overview of the PMC Recipient module.

- **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**
  a. Check to install. Report the following: Pre- and post-installation instructions, customized components, and missing dependent solutions.
  b. Install the solutions: Store the software components in the data dictionary, report additional post-installation instructions.
- **Additional recipient functionality**
  - Uninstall
  - Compare installed solutions
  - Solution History
  - View installation runs.

PMC recipient procedure (CE)

The PMC recipient procedure contains:

- Setup
- To install updates
- Miscellaneous topics

Setup

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

The setup consists of these sections:

- Parameters
- Update VRCs

Parameters

For details see the Parameters (ttpmc0100s000) session.

Note

Because parameter definition is a one-time step, you must be aware of the impact 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.

Hint

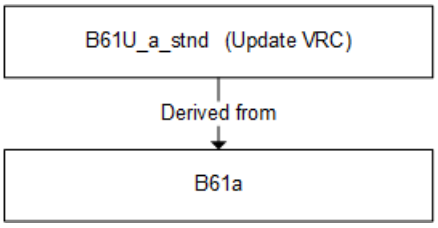
If you have more than one ERP environment running, for example, a Production and a Test environment, you can share the solution dump directories on the ERP environments, which reduces disk-space consumption.

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. The update VRCs are created automatically during the installation of the master product.

In addition, you can define update VRCs using the Update VRC's (ttpmc2140m000) session.

In general, the VRC structure for standard LN resembles this diagram:



B61_a is the VRC that contains the baseline, which is the very 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	

To install updates

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

The procedure consists of:

- Procedure steps
- Scan updates
- Check to install
- Install
- Check to uninstall
- Uninstall

Procedure steps

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

The following sections provide a detailed explanation of the procedure.

To install updates:

1. Scan updates.
2. Check update to install.

3. Install update.

To uninstall updates:

1. Check update to uninstall.
2. Uninstall update.

Scan updates

The first step to make the solutions available to the PMC registry is to scan the solution files by using 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.

Check to install

All the following functionality is controlled by the Process Solutions (ttpmc2101m000) session.

The solutions with the status **Available** that are connected to the update VRCs must be checked with the 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 **Installed**.

Check to uninstall

All the following functionality is controlled by the Process Solutions (ttpmc2101m000) 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 if you want 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) session, PMC recognizes this situation and 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.

PMC recipient session summary (CE)

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.
Update VRC's (ttpmc2140m000)	Maintain the update VRCs. This session registers the update VRCs to which solutions can be connected.
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.
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 in-

	stalled. 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 multi-level. 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.
PMC Cleanup (ttpmc2220m000)	Clean up PMC. This process helps save disk space.
Print Solution/Patch by Update VRC (ttpmc2401m000)	Print detailed information of solutions and patches in 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.
Solution History (ttpmc2560m000)	Display the history of the changes of a solution or patch in an update VRC.

PMC distributor procedure (CE)

This chapter describes the PMC distributor procedures. This chapter contains the following sections:

- Setup
- To create updates
- Feature Pack development
- Customizations

Setup

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

The setup consists of these sections:

- Setup procedure
- Parameters
- Base VRCs
- Base VRC combinations

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.

To create updates

PMC distributor procedures to create software updates:

To create solutions

To create solutions

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. 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.

3. Define dependencies between the solutions.

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

4. 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.

5. Export the solution.

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

6. 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.

7. 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.

PMC distributor session summary (CE)

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.
Solutions (ttpmc1100m000)	Maintain solutions.
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.
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.
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.
Print Solution/Patch (ttpmc1400m000)	Print solutions and patches.
Validate Solution (ttpmc1404s000)	Validate a solution.
Print Components by Solution (ttpmc1420m000)	Print components 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.
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.
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.

Chapter 14
Application Upgrades

14

Introduction (CE)

This section describes the upgrade procedure of the LN application within a multi-tenant cloud environment.

The upgrades, installed in multiple package combinations, can be distinguished in these version types:

- Major

  This is a major LN release, typically once every 18 months. This release may include major data model changes. Data upgrading, through the DUE, may also be part of the version upgrading process. A new version of the DEM content pack is part of the release.

- Minor

  This is an in-between major versions release. Typically, five minor versions are issued following a major release, once every three months. The release may include data model changes which may necessitate data upgrading through the DUE.

- Patch

  Hot-fix or high-priority solution, together with its dependencies to earlier solutions, issued for a major, or minor release. A new patch version does not include any data model changes, and does not result in the release of a new DEM content pack.

You can handle the complete upgrade process of all version types, including the preparation and post-install tasks, in the Install Upgrades and Patches (ttmtm3500m000) session.

Procedure (CE)

The LN application upgrade procedure consists of these phases:

- A preparation phase in which users are informed and the environment is prepared for an upgrade
- The actual installation of the upgrade
- Post installation actions

Note

- For all steps you must use an account with super user authorization.
- A default text group template with a text group for, at least, company 0 is required for this account. To ensure this template is linked to the user account, use the User Data (ttaad2500m000) session. The data must be available on runtime.

Step 1: Preparation

1. Start the Install Upgrades and Patches (ttmtm3500m000) session.
2. Inform users that an upgrade will be done. To define and send the announcement to the users, click **System Message**. For detailed information about system messages, see the online help of the System Messages (ttaad3290m000) session.
3. During an upgrade, normal users must not be able to log on or have connections. To accomplish this, you must switch the environment to Admin Mode: click **Admin Only** and confirm this question: `Environment will be set to Admin Mode. Do you want to continue?`.

If the switch to Admin Mode succeeded, the preparation phase is completed. You can now start the actual installation of the upgrade.

Note: When you log off, or when you are being logged off, the shared memory is shut down. When the first LN user then logs on to a certain node, shared memory is restarted on that node. This may take a few minutes.

Step 2: Upgrade Installation

1. In the **Available Upgrades** field, select the upgrade to be installed. Be aware of the type and consequences of the selected upgrade.

   See *Introduction (CE) (p. 101)*.
2. To start the upgrade process, click **Upgrade** and confirm this question: `Environment will be updated from package combination <source> to <target> (major|minor|patch). Do you want to continue?`.
3. If the upgrade succeeded, perform a restart to actualize the changes and finalize the upgrade installation.

Step 3: Post Install Tasks

1. Depending on your ELN version, perform one or more of these actions:
   - [ELN-3]: If the Extensibility feature is enabled, use the General Table Maintenance (ttaad4100) session to replace the value of the following fields with `EXT<new package combination>`:
     - The **Application Name** field of the Packages by Application table (ttadv121.anam)
     - The **Application Name** field of the Languages by Application table (ttadv125.anam)
     - The **Application Server** field of the Projects table (tlprj110.apps)
   - [ELN-3|4|5]: Ensure the **Display Device type** as defined within the WebUI Print Settings (ttaad3108m000) session is set to PDF output.
   - If LN Studio is used, the activities must be removed from the Activity Explorer. Select the **Delete project contents on disk (cannot be undone)** check box. After deleting the activities from the workspace, they can be opened again.
2. To check whether customer data must be updated after the application upgrade, click **Data Upgrade Engine**. If customer data must be updated, you must perform a data upgrade run. To perform this step, confirm this question: `Data Upgrade Run for package combination <target> will be executed. Do you want to continue?`.

   When you confirm that you want to continue, the data upgrade run is started automatically. Alternatively, you can handle the run manually through the Data Upgrade Runs (ttspt2500m000) session.

   For detailed information about the Data Upgrade Engine, see these sections:
   - *Application Upgrades (p. 101)*
   - The online help of the Data Upgrade Runs (ttspt2500m000) session
3. After the upgrade installation and the, optional, data upgrade has been performed successfully, allow the normal users access to the environment again. To perform this step, click **Bring Online** and confirm this question: `Environment will be set to Online. Do you want to continue?`

Chapter 15
Data Upgrade Engine

15

Data Upgrade Engine overview (CE)

The Data Upgrade Engine (DUE) is used to update the customer data after a major or minor Application 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 following example.

Example

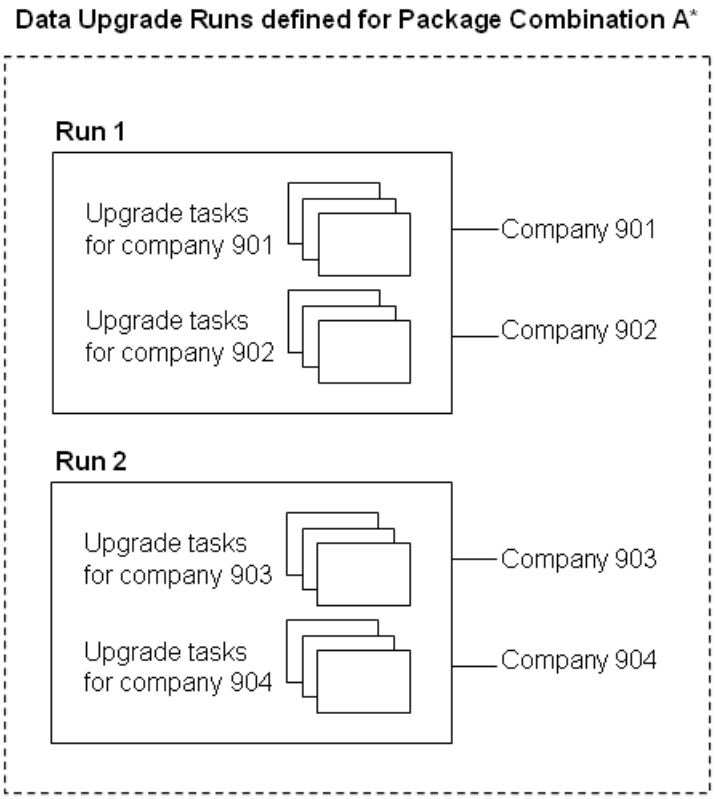
These companies are linked to package combination A:

- 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:



* Package combination A is linked to companies 901, 902, 903, 904

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 have been executed.

3. Runtime Class

   If a batch contains multiple upgrade programs whose pre-requisites have been 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.

For details on how to create and execute a data upgrade run, see Executing a data upgrade run (OP-CE).

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 (OP).

Troubleshooting

If one or more upgrade programs fail during a data upgrade run, use the troubleshooting procedure. See *Troubleshooting (OP-CE) (p. 115)*.

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 (CE)

This section describes how to execute a Data Upgrade Run.

To execute a data upgrade run:

Step 1: Display the run information

1. Start the Data Upgrade Runs (ttspt2500m000) session.
2. Go to the **Run Information** tab.

Step 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:

1. Select the run and on the appropriate menu, select **Initialize Data Upgrade Run**. The Initialize Data Upgrade Run (ttspt2200m000) session starts.
2. 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 will 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 filled in. The data upgrades to be performed will be based on the specified source feature pack number:
   - If the number is too low, too much data is converted. Possibly data will be overwritten with wrong values.
   - If the number is too high, not all data is converted. Possibly data will not be 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.
3. Click **Initialize**.
4. When the initialization is finished, a "Process completed" message is displayed. To remove the message box, click **OK**.
5. Close the Initialize Data Upgrade Run (ttspt2200m000) session.

Step 3: 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:

1. In the Companies by Data Upgrade Run (ttspt2510m000) session, select the companies you want to move.
2. On the appropriate menu, select **Move Companies to Another Run**. The Move Companies to Another Run (ttspt2210m000) session starts.
3. Enter the required information:
   - Optionally, change the company selection.
   - Specify the destination run. You can create a new run, or select an existing run.
4. 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 (OP-CE) (p. 115)*.
- If you set the **Real Run Status** to **Released**, the task will not be 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.

Step 4: Execute the run

1. Select the run in the Data Upgrade Runs (ttspt2500m000) session.
2. On the appropriate menu, select **Data Upgrade Engine**. The Data Upgrade Engine (ttspt2201m000) session starts.
3. 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 (OP-CE) (p. 115)*.

Performance and finetuning (CE)

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.
- The usage of additional servers. See Additional servers.
- The usage of the local server for processing. See Using the local server for processing.
- The usage of sub-tasks. See Using sub-tasks.

You can use the Call Graph Profiler to identify potential performance bottlenecks. See Using the Call Graph Profiler.

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 may be factors that lead to a not optimal determination of the runtime class. As a result long running tasks may 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 (OP-CE).

   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 will 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, it is important to investigate the maximum duration of an upgrade task before you change the runtime class.

Additional servers

In the **Number of Additional Servers** field in the Data Upgrade Engine (ttspt2201m000) session, you can specify the number of additional bshells that will be used to run the Data Upgrade Engine.

Additional bshells can greatly improve the performance of the DUE because several upgrade tasks can be started in parallel.

The maximum number of additional servers that can be used is restricted by the Infor Cloud team.

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 will not only schedule the upgrade tasks but will also run upgrade tasks.

We recommend the following:

- 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 **Profile Mode** 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 (OP-CE)

Procedure

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 and 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 will be written to the `$BSE/log/log.due` file. This 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 the "Performance and finetuning" section in this guide. Performance and finetuning (OP).

Chapter 16
Application Personalization

16

Introduction (CE)

You can personalize the application to your own preferences. Ensure to select the 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.

- Session Personalization (ttadv9403m000)

  Use this session to view and export or import session personalizations. Each record in the grid represents one or more personalizations.

- Menu Personalizations (ttadv9505m000)

  To maintain the menu personalizations for LN UI users.

- Export Configuration Data (ttaad7201m000)

  To export miscellaneous types of configuration data.

- Import Configuration Data (ttaad7202m000)

  To import miscellaneous types of configuration data as exported by the Export Configuration Data (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 (OP-CE)

You can define and apply personalizations at these levels:

- User
- Role
- DEM roles
- Company

The created personalizations at role level, not to be confused with DEM roles, are designed to function in a non-DEM 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 Current Role (ttdsk2009m000)

Menu personalization (OP-CE)

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 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 *Infor Ming.le-LN Plug-in - User Guide (LN UI)*

Chapter 17
LN Studio

17

Introduction (CE)

LN Studio is a development platform for LN and is implemented in the Eclipse framework. The *Infor LN Studio Administration Guide* contains more information about LN Studio. That document is applicable for use on premises. In the cloud, you can use LN Studio only to edit and debug components that are used for extensions.

The system administrator has these tasks related to LN Studio:

- Register LN Studio as authorized app in ION API.
- Install LN Studio.
- Set up the development environment.
- Make LN Studio available for all users.
- Make the connection settings available for all users.
- Publish updates for LN Studio.

Prerequisites (CE)

Licenses

Adapter for LN – product ID 7056

Client PC

These are the prerequisites for the client PC:

- 2 GB RAM memory, recommended
- Microsoft Windows 7 or later

- Java Runtime Environment (JRE) 1.7(Java 7) or later, 64-bits version
  You can download the JRE software from http://java.com/download .

Registering LN Studio as authorized app (CE)

To perform the registration, use the ION API application within the Infor Ming.le ™ portal. See the *Infor ION API Administration Guide*.

1. Add a non- Infor authorized app of the Web Application type.
2. Specify this information:
   - Description: Specify LN Studio.
   - Redirect URL: Specify `oob://localhost/lnstudio`.
   - Authorized Javascript Origins: Specify `http://localhost`.
3. Download the credentials.

This results in a file with the `.ionapi` extension. All LN Studio users require this file to create a connection to the cloud.

Installing LN Studio (CE)

Step 1: Download the zip file with the LN Studio software

Download the `InforLNStudio_10.6.0.nnn-x86_64.zip` file from KB 1561993 on https://www.inforxtreme.com . “nnn” represents a build number, such as “0317”.

Step 2: Install LN Studio

1. Create an installation folder on the client PC. The recommended installation folder is `C:\Infor\LN\LNStudio`.
   Do not install in a Windows protected folder such as `C:\Program Files or C:\Program Files (x86)`.
2. Extract all contents of the `InforLNStudio_10.6.0.nnn-x86_64.zip` file to the new installation folder.

Step 3: Define connection points

To define a connection point for Project Server:

1. Start LN Studio by running `eclipse.exe` in the installation folder.

2. In the **Workspace Launcher** dialog box, select a location for the workspace. Perform one of these actions:
   - Accept the default - inside your “user home” folder.
   - Browse to an existing folder.
   - Specify a new folder to store your work.

   Then click **OK**.
3. When LN Studio has started, select **Window > Preferences**.
4. Add a cloud environment:
   a. In the tree in the left pane of the **Preferences** page, select **Infor LN JCA Connectivity > Cloud Environments**.
   b. To add a cloud environment, click **Add**.
   c. Select **Import from file** and browse to the location of the `.ionapi` file that was created when registering the authorized app in ION. To complete the browse dialog box, click **Open**.
   d. Click **Import**.
   e. Copy the value of the **Gateway URL** to the clipboard.
   f. Click **Finish**.
5. Add a server certificate to the trust store:
   a. In the tree in the left pane of the **Preferences** page, select **Infor LN JCA Connectivity > Trusted SSL Certificates**.
   b. Click **Browse** to select the folder to place your trust store that contains certificates for trusted HTTPS connections.
   c. Select an existing trust store or specify the name for a new trust store. For example, `LNStudioTrustStore.jks`.
   d. To complete the browse dialog box, click **Open**.
   e. Optionally, specify a password for the trust store. If you leave the **TrustStore password** field blank, a default password is used.
   f. Click **Apply**. If the trust store did not yet exist, you are prompted whether you want to create a new one. Click **Yes**.
   g. In the **URL to be checked** field, paste the **Gateway URL** that you copied earlier.
   h. Click **Check**. Usually, a "Check failed" dialog box, which suggests to import the certificate presented by the server, is displayed. Click **Yes**. The **Import Certificate** dialog box is displayed.
   i. In the **Import Certificate** dialog box, the **Server Certificates** drop-down list may contain more than one certificate. The HTTPS server presents a chain of certificates, starting with its own certificate. Next is the certificate of the Certification Authority (CA) that issued the server certificate, followed by the issuer of the CA certificate, and so on. The last certificate in the chain is usually a "root CA certificate", issued by itself. You must import the last certificate. This certificate is already selected in the drop-down list. Details of the selected certificate are displayed in the dialog box.

j. In the **Alias** field, specify a short name for the certificate. This alias is stored in lower case in the trust store. Click **Finish**. The certificate is now present in the **Certificates** table.

k. Click **Check** again. A "Check succeeded" dialog box is displayed.

6. Configure the Project Server connection point:

a. In the tree in the left pane of the **Preferences** page, select **Infor LN JCA Connectivity**. A table with connection points is displayed.

b. Select `ProjectServer` and click **Edit**. The **Configure Connection point** wizard starts.

c. In the **Activation Type** drop-down list, select `HTTP-OAuth2` and click **Next**.

d. Select the correct **Cloud Environment** from the drop-down list.

e. Ensure the **Company** field is set to 0.

f. Click **Finish** to end the **Configure Connection** wizard.

g. Ensure the `ProjectServer` connection point is selected, and click **Ping**.

h. A browser dialog box with a login screen is displayed. This dialog box is presented by the authorization server in the cloud. Specify your credentials for accessing the cloud.

i. You are prompted whether you want to grant application access to the LN Studio. Click **Allow**.

j. A “Ping succeeded” message is displayed. Click **OK**.

This validates that the connection to the cloud can be made.

Setting up the development environment (CE)

Step 1: Generate the development environment

1. Open a browser and start the LN application (LN UI).
2. Start the Extensions (ttext1500m000) session.

   A development environment is now automatically generated. You do not have to manually create a development environment in the **Project Server** perspective of LN Studio.

Step 2: Verify the development environment was generated successfully

1. Start LN Studio.
2. Open the **Project Server** perspective. Select **Window > Perspective > Open Perspective > Project Server**.
3. A browser dialog box with a login screen is displayed. This dialog box is not displayed if you have logged in earlier in the same LN Studio. Specify your credentials for accessing the cloud. On the next **Request for Approval** dialog box, click **Allow**.
4. In the **Application Explorer** view on the right, a development environment named “EXT” is displayed. Expand the “EXT” node by clicking on the **>** in front of it.

5. Expand the **Applications** node. You are prompted to create the required Administrator connection. Click **Yes**.
6. In the **Configure Connection point** wizard, select **Share connection**. In the **Shared Connectionpoint** list, select **ProjectServer**. Click **Finish**.
7. After the Administrator connection has been configured, the **Applications** node is expanded to show an application with a name such as “EXT1052002”.
8. Right-click the application and select **Properties**.
9. In the **Application properties** dialog box, verify that the **Release** has been filled. Also verify that the **Packages** table only contains the “Infor LN Extensions” (tx) package. Click **Cancel** to close the dialog box.
10. In the **Software Project Explorer** view on the left, a software project with the same name as the application is displayed. Right-click this project and select **Properties**.
11. In the **Project properties** dialog box, verify that the **Status** is “Active”. Also verify that the **Development Environment** and **Application** match the values in the **Application Explorer**. Click **Cancel** to close the dialog box.

Making LN Studio available for all users (CE)

1. Download the `InforLNStudio_10.6.0.nnn-x86_64.zip` file from KB 1561993 on https://www.inforxtreme.com . “nnn” represents a build number, such as “0317”.
2. Publish this zip file on a location that is accessible by all users of LN Studio, such as a file share on the local network.

Making connection settings available for all users (CE)

The `.ionapi` file that was downloaded during the registration of LN Studio in ION API must be distributed to all LN Studio users. You can email this file to all users.

Note

This file contains sensitive data. If you publish this file on a file share in the local network, ensure that unauthorized users do not have access to this file.

Publishing updates for LN Studio (CE)

If an update for LN Studio is available, download the updated zip file from KB 1561993 on https://www.inforxtreme.com . Publish the new zip file in the same way as the original zip file. Notify all users that a new version is available.

Chapter 18
Miscellaneous

18

Quick Flow (OP-CE)

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.
- 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, for using quick flow/message suppression is documented in these guides:

- *Infor Ming.le-LN Plug-in - User Guide (LN UI)*
- *Infor Ming.le-LN Plug-in - User Guide (Web UI) (U9647)*

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 linked to the modified template are now authorized to suppress messages and remember question selections.

MS Excel integration (CE)

Introduction

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.
- 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) (U9647)*

System Messages (CE)

To send system information to users and administrators, you can use system messages. The System Messages (ttaad3290m000) session is created to easily send messages to specific users and administrators on the server.

The super user 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.

Appendix A
Glossary

A

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.

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.

BOL

See: *Business Object Layer (p. 129)*

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

Acronym: BOL

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.

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.

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 Upgrade Engine

A framework that executes application data upgrade programs.

The Data Upgrade Engine is used after installation of a Feature Pack.

Acronym: DUE

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.

DUE

See: *Data Upgrade Engine (p. 130)*

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.

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.

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.

PMC

See: *Product Maintenance and Control (p. 132)*

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.

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.

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.

Acronym: PMC

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.

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.

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.

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.

Index