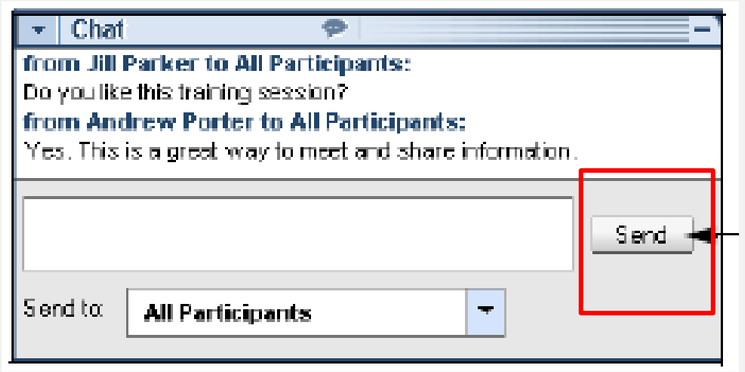

infor
TM

Education

Landmark Technology: Using Landmark Pattern Language (LPL)

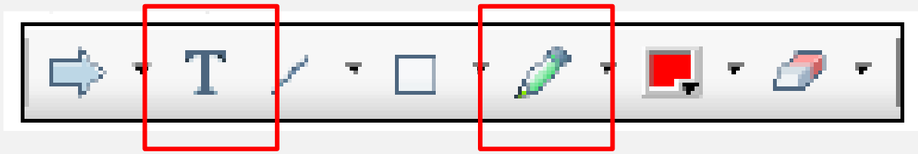
Tools you can use

- Chat
  - Type the question and click Send.



- Raise your hand
  - Let the facilitator know you have a question.

- Annotation tools
  - Select an annotation tool and click inside the Content Viewer to use.



Introductions

- Name
- Location
- Role
- Course expectations
  - What course topics are you most interested in?
  - How are you planning to apply this class to your day-to-day role?

Course exercises

- Exercises and demos section
  - Located at the end of the training workbook
  - Contains step-by-step instructions for all exercises
  - Each exercise is hyperlinked for easy navigation
    - Jump between the lesson content and the exercises
  - Print exercise pages without having to print the entire workbook
- Exercises are built to be performed on a training server
  - Access the training server using the link provided by your instructor
  - Each participant receives access to an individual training environment
  - Log in through this virtual machine: **LMK11 (Landing Server) VM**

Course overview

Learning objectives

- After completing this course, you will be able to:
  - Describe Landmark Pattern Language (LPL).
  - Describe Landmark ontology.
  - Configure a business class.
  - Create a user business class.
  - Create user fields.
  - Configure common user interface objects.
  - Configure an action.

Course agenda

- Lesson 1: Overview of Landmark Pattern Language
- Lesson 2: Ontology and object relationships
- Lesson 3: Configuring business classes
- Lesson 4: User business classes
- Lesson 5: User fields
- Lesson 6: Configuring user interface objects
- Lesson 7: Configuring actions

Appendices

Appendix	Appendix title	Content description
A	User accounts	A reference for student and instructor login credentials
B	Access the LPL reference guide	Step-by-step instructions for accessing the LPL reference guide
C	Scripts and script engine	Instructions for adding scripts to Landmark
D	LPL code for course exercises	LPL code that can be used while performing course exercises

Note: A section called **Exercises and demos** containing detailed instructions for all course activities appears immediately after the appendices.

Lesson 1: Overview of Landmark Pattern Language

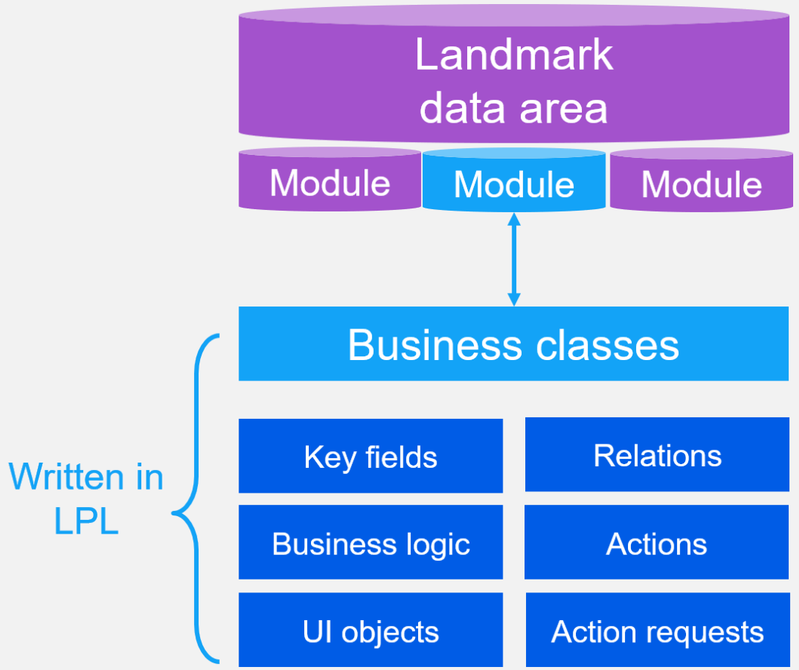
Learning objectives

After completing this lesson, you will be able to describe Landmark Pattern Language (LPL). In this lesson, you will:

- Define the purpose of LPL.
- Discuss using LPL for application configurations.
- Identify ways to access LPL.
- Describe the basic syntax of LPL.

About LPL

- Object-oriented language
- Used to define data and logic within Landmark applications
- Many source files are written in LPL
- LPL code in a business class defines characteristics such as:
  - What data is stored in the Landmark data areas
  - What fields are available for forms and lists
  - What actions can be performed on the data



Purpose and benefits

- Purpose of LPL is to enable customer upgrade extensibility by offering a high degree of flexibility and low complexity
- Benefits:
  - Supports rapid construction of powerful applications by eliminating Java programming which is arduous and time consuming
  - Allows Landmark administrators and business analysts to focus on business functionality and processes
  - Increases user productivity of the application due creating tailored user experiences and efficiencies down to the field level
  - Lowers costs because users can manage configurations and application upgrades are easier and generally error free

Who uses LPL?

User categories	Description	Access tool
 General Landmark users	Typical users do not use LPL at all. They access applications from the web interface which, apart from a few exceptions, does not allow access to the LPL. End users who have access to Infor Rich client can use LPL to personalize lists or forms.	• Web UI • Infor Rich Client via the Options > Personalize menu on a list or form or Ctrl-Shift-Click
 System administrators and power users	Users with specialized security can access LPL to configure delivered business objects and UI objects, and create new objects specific to a customer's requirements.	Configuration Console
 Infor developers and business analysts	Developing and maintaining delivered business classes and user interface definitions - This involves changing the LPL source code.	Landmark Application Designer IDE

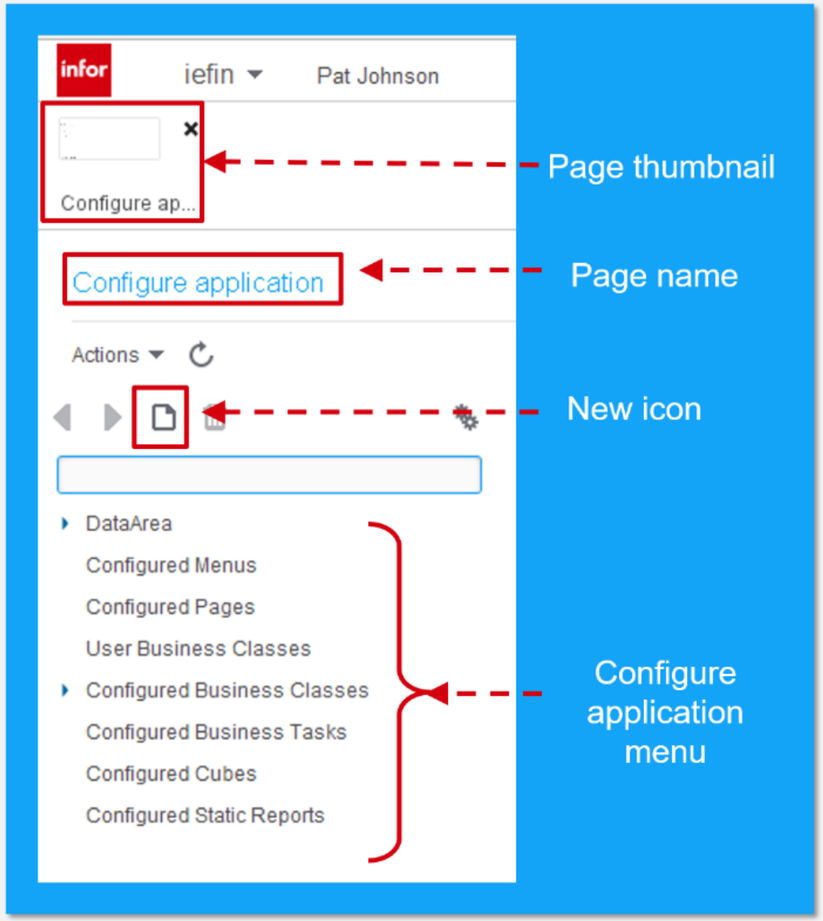
Configuration Console

Applications	• Configure business classes for these purposes:
	  ○ Adapt applications to support organizational processes
	  ○ Modify the look and feel of application screens
	• Configure features that apply across a data area such as data translations, time zones, business subjects, scripts, phrases and MIME types

Security	• Create security and modify security classes • Maintain actor, identity, and role records
Web services	• Enable non-Landmark systems to view or update Landmark data

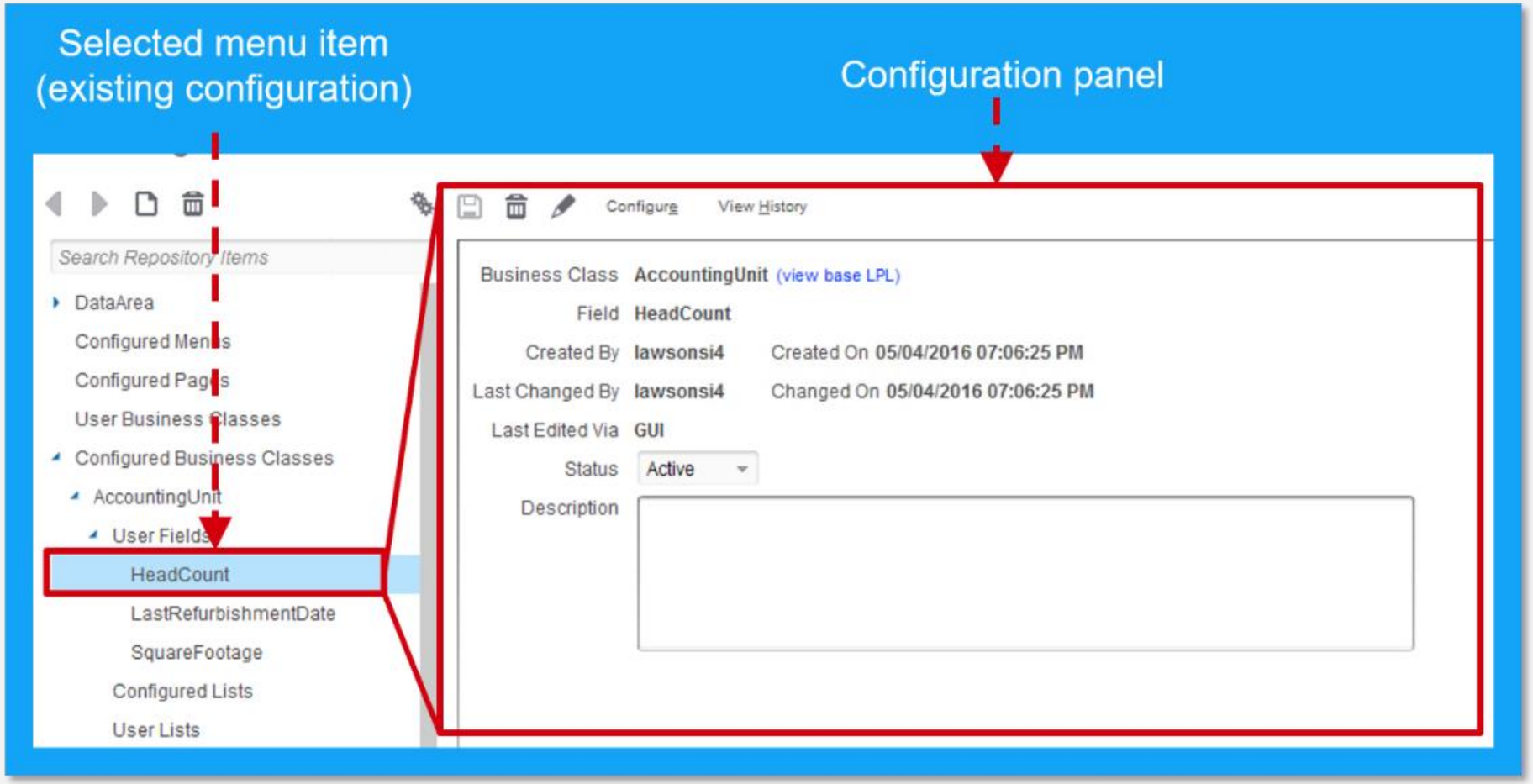
Configuration Console, continued

Configure Application page – left panel



Configuration Console, continued

Configure Application page – right panel



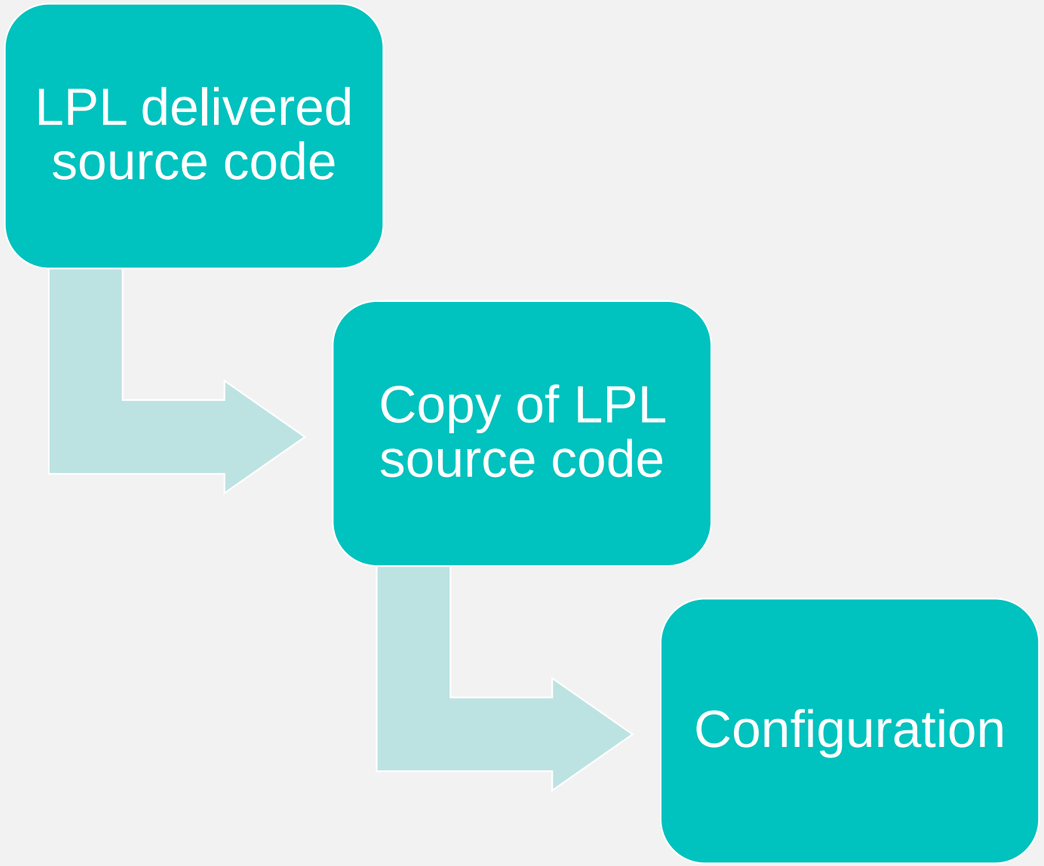
About application configurations

- Global changes that affect the experience of those who work with modules
- Change what users see on screen and the tasks they can perform
- Modify components such as:
  - Menus
  - Pages
  - Forms
  - Fields
  - User interface objects
- Can be made to existing components or can create new components.
- Changes are done in real time, which means that changes take place immediately in a running system with no down time

About application configurations, continued

LPL source code changes

- A configuration changes a copy of the LPL source code, not the original delivered LPL source code.
- The LPL source code is never accessible or modifiable by the customer.
- The configured LPL code is stored separately in the database.
- Configured changes apply to all users.



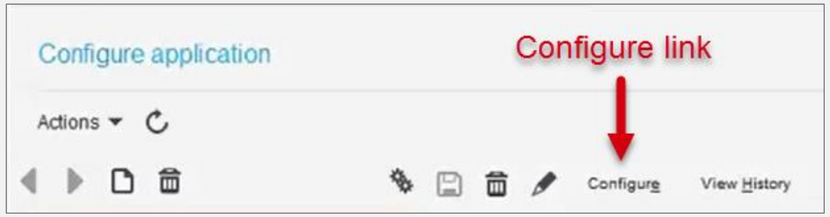
Accessing LPL

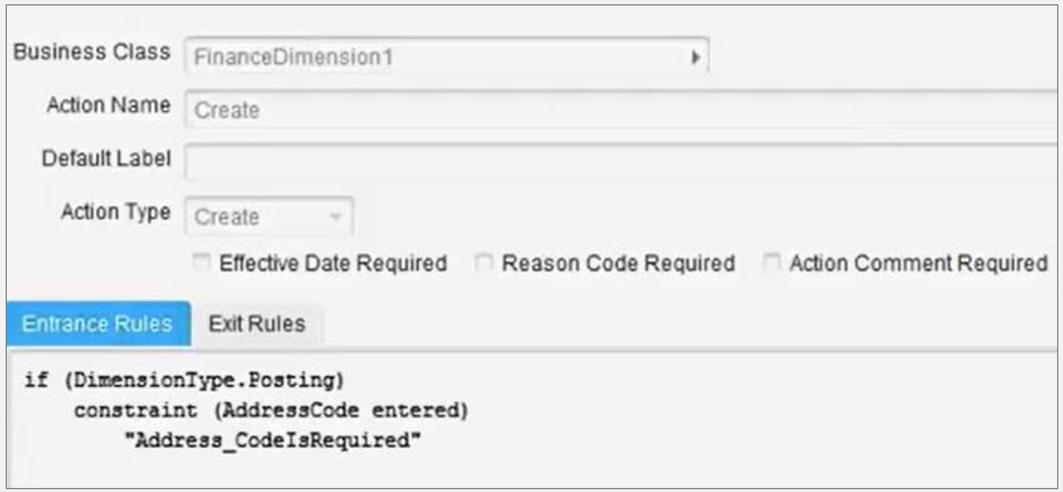
- Here are the primary ways to access LPL code to configure an application:
  - Configure link
  - LPL Editor (pencil icon)
  - LPL Viewer
- Another way to access LPL code is through the View base LPL link found on certain configurations.

Accessing LPL, continued

Configure link

- Accessible from the right panel of Configuration Console
  - Existing configuration must be open
- Click the Configure link to open the Configure form.
- Gives partial access to LPL code in limited areas such as:
  - Entrance or exit rules
  - Persistent or derived fields
- You may need to open the full LPL Editor.



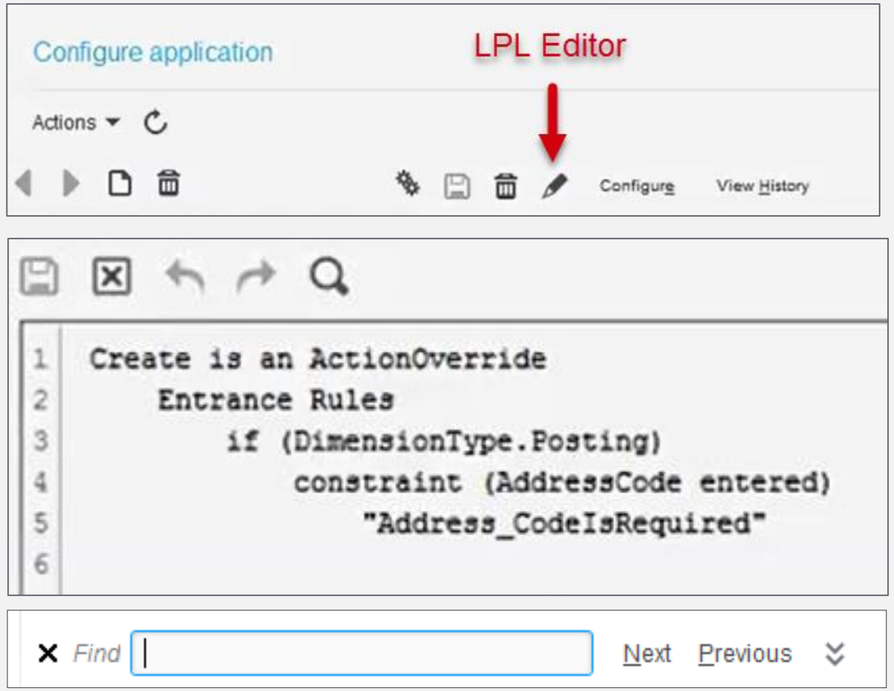


Configure form example

Accessing LPL, continued

LPL Editor

- Referred to as the pencil
- Accessible from the right panel of Configuration Console
  - Existing configuration must be open
- Click the pencil to open the LPL Editor.
- Gives full access to LPL code
- Ctrl+F opens a search box at bottom of LPL Editor

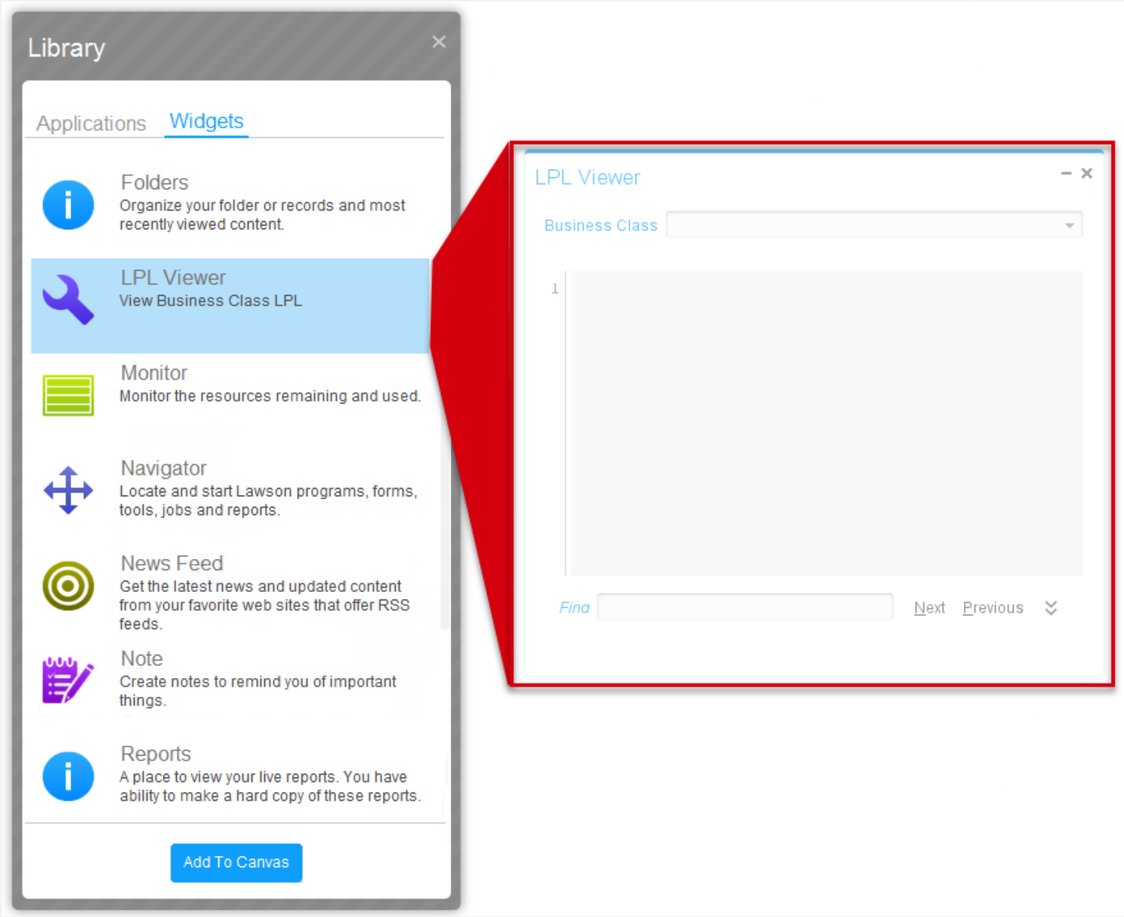


LPL Editor example

Accessing LPL, continued

LPL Viewer

- Window into LPL code that is outside of Configuration Console
- Select Start > Library > Widgets > LPL viewer.
- View LPL for any business class.
  - Select a business class from the drop-down list in the LPL Viewer.
- View only, cannot edit LPL code



Exercise 1.1: Log in to Infor Rich Client and explore Configuration Console

- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Accessing Web Configuration Console

- The Web Configuration Console is a tool that enables administrators and power users to make changes that affect the user experience and runtime logic of Landmark-based applications. *(Release Date to be announced)*

- Web Configuration Console for the web is only available in SoHo 4 UX.

- *Note: The Web Security Administration application will also be available in SoHo UX to create security configurations.*

Exercise 1.2: Explorer the Web Configuration Console

- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Landmark data structure

- LPL files
  - A logical representation of database structure and constraints, plus business logic
  - Describe a collection of business classes, business tasks, modules, fields, and other miscellaneous files
- A runtime dictionary is generated from these files that includes all the structural metadata for the persistent portion of the business classes.

Introduction to LPL syntax

- Syntax refers to the set of rules that define how words and symbols can be combined into strings that are understood.
- LPL syntax uses an adaptation of Backus-Naur Form (BNF) which:
  - Defines the parts of the language and related applicable rules.
  - Is designed to be concise, precise, and easy to learn and understand.
- The full complexity of LPL syntax is beyond the scope of this course.

For your reference

For details on LPL syntax, please see the Configuration Console Reference Guide: Landmark Pattern Language (LPL) and the Infor Java Framework Declarative Metadata document.

Introduction to LPL syntax, continued

Text

- All LPL text requires:
  - Concatenate multiple word phrases
  - No spaces
  - No hyphens
  - Capitalize first letter of each word
- Text messages additionally :
  - Always contained between quotes
  - Only the first word of the message will display with an uppercase letter.
    - If you put an underscore before a word, it will also display as uppercase.

```
Delete is an Action
    Action Rules
        constraint (ProjectEmployeeRel not exists)
            "CannotDelete;ProjectLaborScheduleUsedByEmployees"
```

Introduction to LPL syntax, continued

Block structure and local line up

- Tab-oriented
  - Each tab = four spaces
  - Line of code must be tabbed in if subordinate to the line above it
- Block structure
  - Denoted by indent level
  - Block used even for one statement
  - Blank line after end of block
- Local line up
  - Provides visual clarity
  - Helps with quick comprehension

```
1  ProjectLaborSchedule is a BusinessClass
2      owned by Projects
3      prefix is PJLSC
4
5      Ontology
6          symbolic key is ProjectLaborSchedule
7
8      Persistent Fields
9          Description
10         Active
11         FirstPeriodStartDate    is Date
12         Frequency               is Numeric size 1
13             States
14                 Weekly          value is 1
15                 BiWeekly        value is 2
16                 SemiMonthly     value is 3
17                 Monthly         value is 4
18                 FourWeekly      value is 5
19
```

Introduction to LPL syntax, continued

Keywords and conditional statements

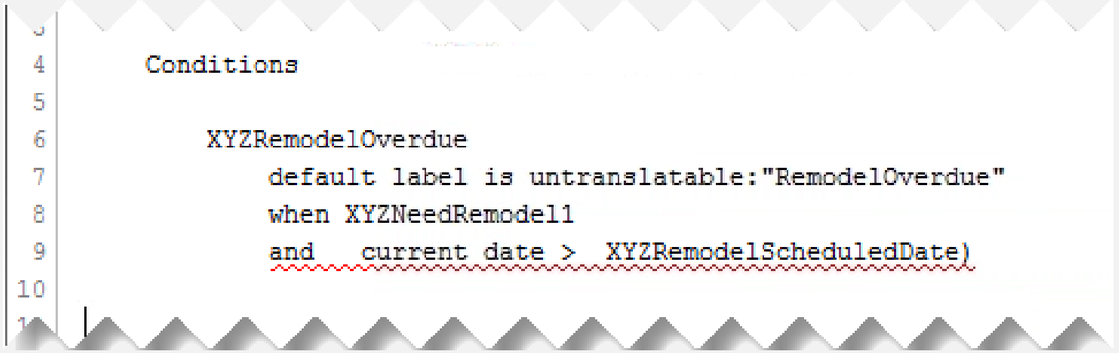
- LPL uses common programmatic keywords:
  - While
  - When
  - If and Else
- Location of conditional keywords
  - Beginning of next line
  - Indent based on logical structure of the conditional statement

```
26
27      Derived Fields
28          SalaryDivisor is a DerivedField
29              type is Numeric size 2
30              restricted
31              if (Frequency.Weekly)
32                  return 52
33              else
34              if (Frequency.BiWeekly)
35                  return 26
36              else
37              if (Frequency.SemiMonthly)
38                  return 24
39              else
40              if (Frequency.Monthly)
41                  return 12
42              else
```

Introduction to LPL syntax, continued

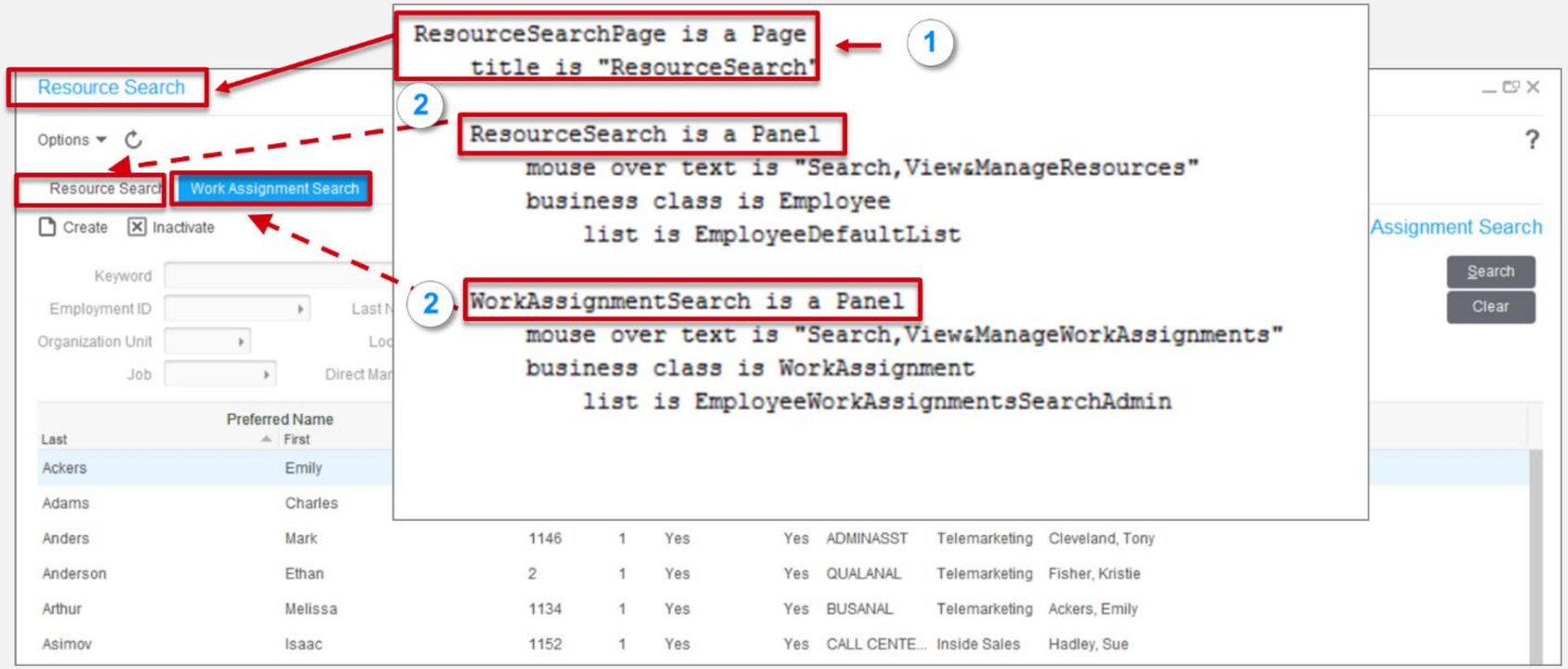
Identifying invalid code

- Easy to identify on screen
- Red squiggly line under invalid code such as:
  - Typing error
  - Invalid business class
  - Unfinished expression
- Appears as you write an unfinished line
- Disappears after you complete the expression



Example of LPL source code

- Example of the page that displays if you select Resources in Global Human Resources (GHR)
  1. The LPL code identifies this as a page and provides a page title, Resource Search.
  2. The LPL code defines the page to displays with two tabs (panels), each with specifications for mouse-over text and what list to display when the tab is selected.



Check your understanding

Check your understanding

Describe Landmark Pattern Language (LPL).

LPL is an object-oriented design language used to define data and logic within Landmark applications.

Many of Landmark’s source files, such as business classes and user interface objects, are written in LPL.

Check your understanding

Name at least four components of a business class that are written in LPL.

Here are six:

- Key fields
- Business logic
- Relations
- Actions
- Action requests
- User interface objects (e.g., forms, fields, and menus)

Check your understanding

Which two of the following statements about application configurations are true?

a) They are made to the original LPL code.
b) They are applied in real time.
c) They affect only the user who made the configuration.
d) They can be made to existing application components or can create new components.

Check your understanding

Describe basic syntax for LPL.

- When writing LPL text, concatenate multiple-word phrases.
- LPL follows a block structure where tabs are important.
- LPL uses common programming keywords such as While, When, If and Else, as well as some LPL-specific keywords.
- Invalid code is easy to identify because a red squiggly line appears under LPL code that is unfinished or not valid.

Lesson 2: Ontology and object relationships

Learning objectives

After completing this lesson, you will be able to describe Landmark ontology. In this lesson, you will:

- Define ontology.
- Describe the Landmark ontology model.
- Explain object relationships.

Ontology concepts

- Ontologies exist in many fields, from information science to philosophy.
- What they all have in common is a goal of representing ideas, with all their interdependent characteristics and relations, according to a system of categories.
- In computer science, ontology is an organizing framework for understanding entities and relationships within a domain.
- Ontologies offer a better way than a relational database to manage data and the complexities of the real world.
  - Ontologies provide the means to store information about what the concepts are and how they relate to other concepts.
  - Because relationships are kept with the data, you can construct advanced queries.

Landmark ontology

- Affordance structure which defines the order in which things must exist to have meaning
- Uses the natural relationships that objects have with one another
- Based are two primary objects: **business classes** and **key fields**
- A business class describes the data model and business logic for a business object. In includes:
  - Persistent fields
  - Derived fields
  - Rules
- A key field defines the symbolic (human-readable) key for a business class.
  - Context is defined in the key field.
  - Context determines what makes the key field unique.

Landmark ontology, continued

- Landmark contains two special business classes:
  - **Enterprise Company** (key field = company)
  - **Enterprise Group** which is a set of companies that share resources such as Vendor, Employee, Customer
- As a standard, all Landmark business objects are directly or indirectly in the context of one of these two business classes.

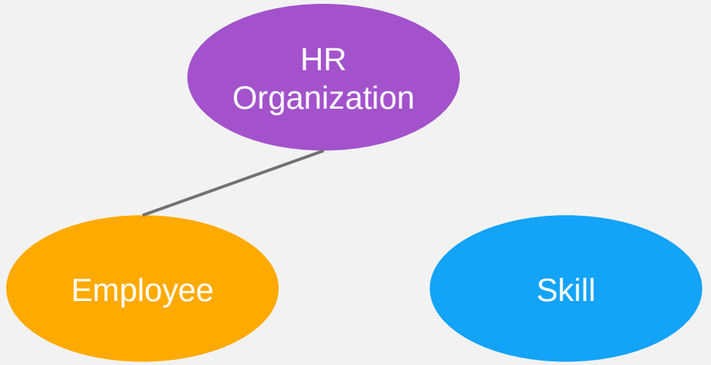
Object relationships

- Example of the relationships between three simple objects
- Objects are for illustrative purposes only
- We use the term HR Organization in the graphics as this is commonly used in Landmark applications such as Global Human Resources (GHR)

Object relationships, continued

Context

- Context defines the objects that afford the existence of another object.
- In the example:
  - You must have a HROrganization to define an employee.
  - A skill like programming does not need to be tied to a HROrganization.
- The LPL code
  - Employee key field defines employees as being represented by a number up to 13 digits.
  - Each employee is unique within the HROrganization context because Employee 1 in HROrganization A is unique from Employee 1 in HROrganization B.



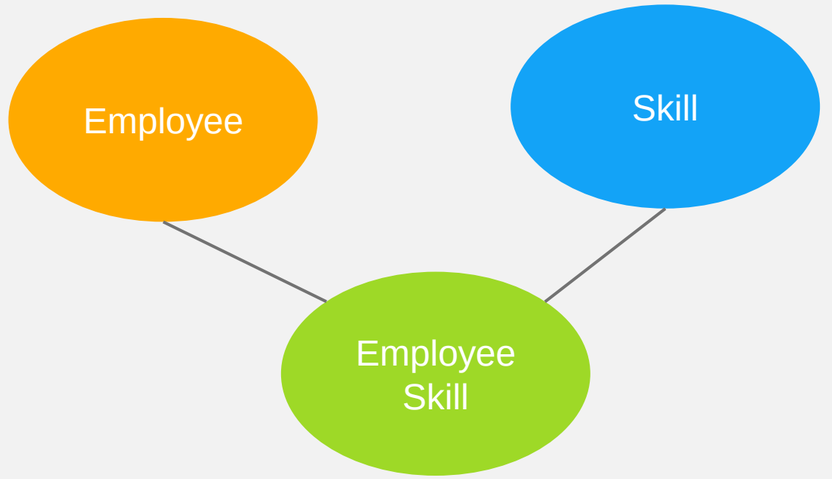

```
Employee is a KeyField

    Representation
        type is Numeric 13

    Ontology
        stereotype is Agent
        business class is Employee
        Context
            HROrganization
```

Object relationships, continued

Dual affordance

- Dual affordance means that an object is in the context of two or more objects.
  - Since Employee is in the context of HROrgnaization, EmployeeSkill is also implicitly in the context of HROrganization.

Example: If you change the context of Employee to Department instead of Company, the context for EmployeeSkill will automatically change to Department.

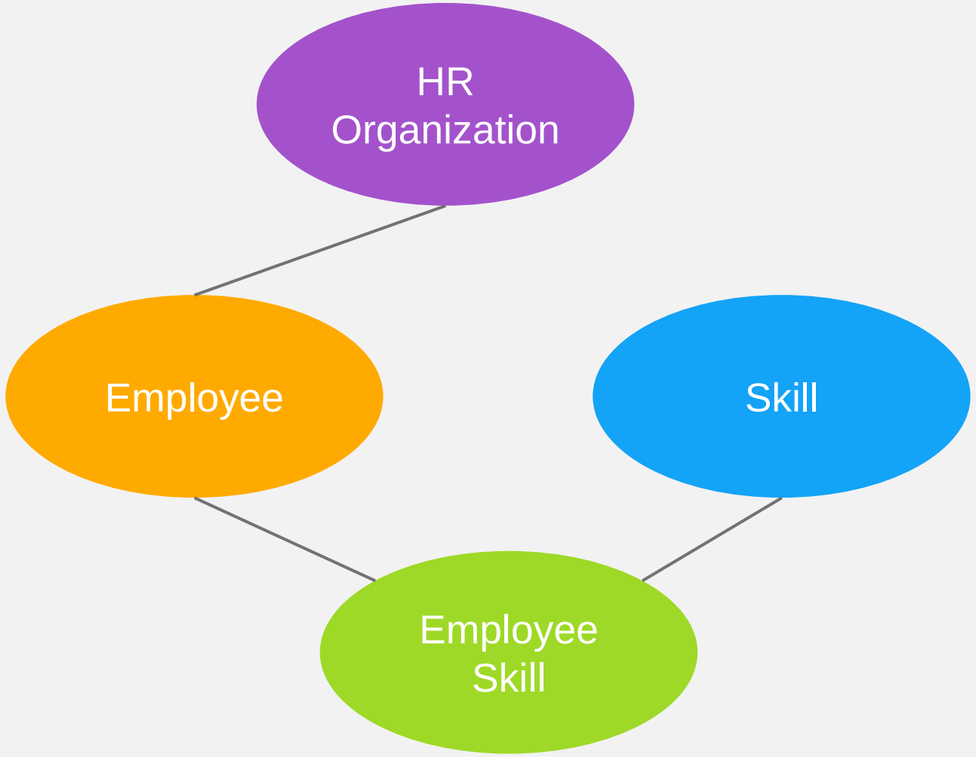


EmployeeSkill **is a KeyField**

Ontology
business class is EmployeeSkill
Context
Employee
Skill

Built-in relationship examples

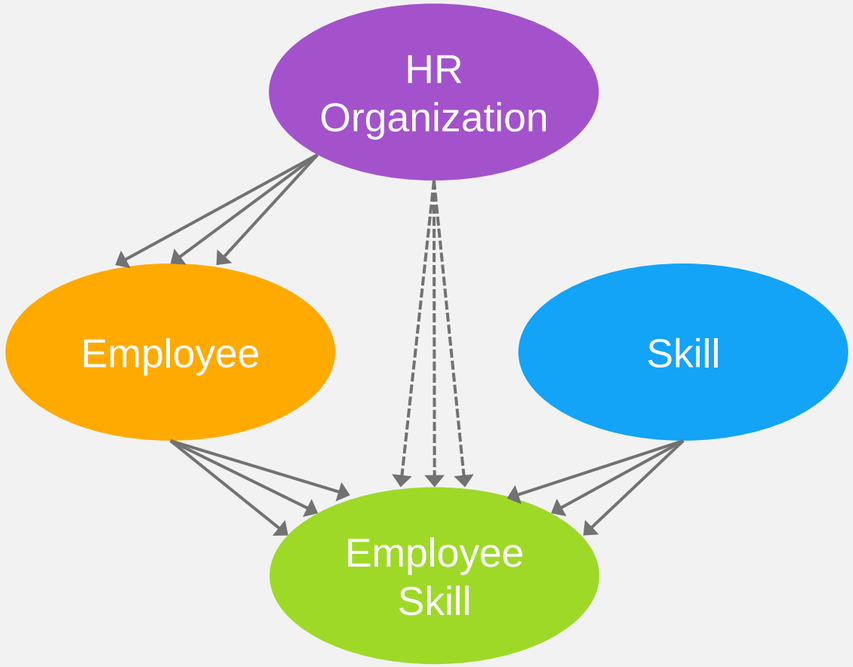
- Some relationships are automatically established when providing context among objects.
- We will look at the automatic relationships that were created from the current example model.



Built-in relationship examples, continued

One-to-many relationships

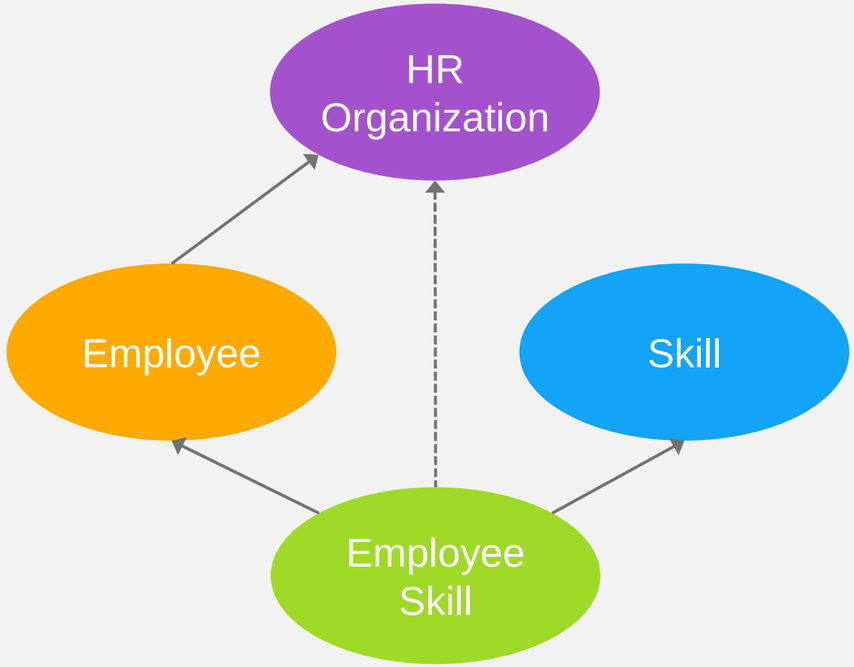
- Mother-to-children concept
  - One mother can have many children.
- In our example:
  - A HROrganization can have many employees.
  - An employee can have many skills.
  - The dotted lines shows the implicit context between HROrganization and Employee skill.



Built-in relationship examples, continued

One-to-one relationships

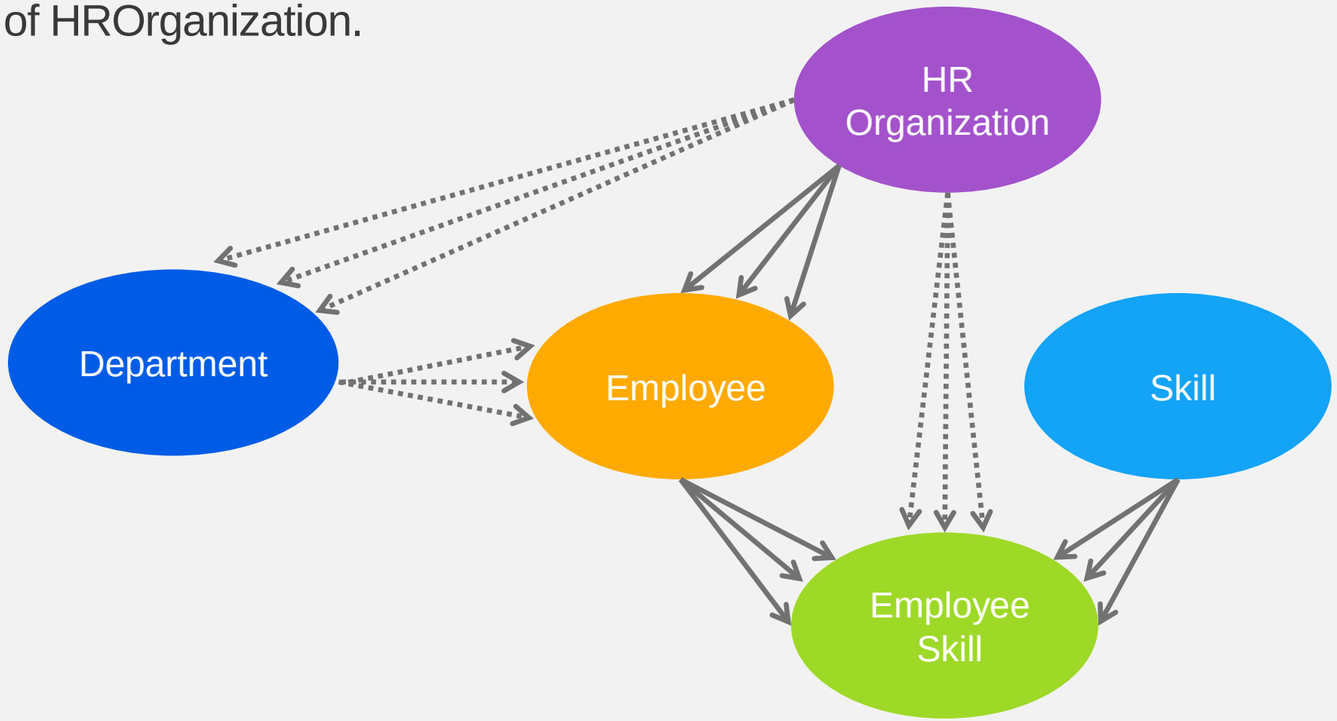
- Child-to-mother concept
  - A child can have only one mother.
- In our example:
  - A HROrganization can have many employees.
  - An employee can have many skills.
- The keyword “name” is how LPL refers to a one-to-one relationship.
  - If you want to find the name of the HROrganization for a particular employee, refer to HROrganization.Name.
  - LPL uses the dot notation in coding to navigate these relationships.



How to use relationships

Add a business object

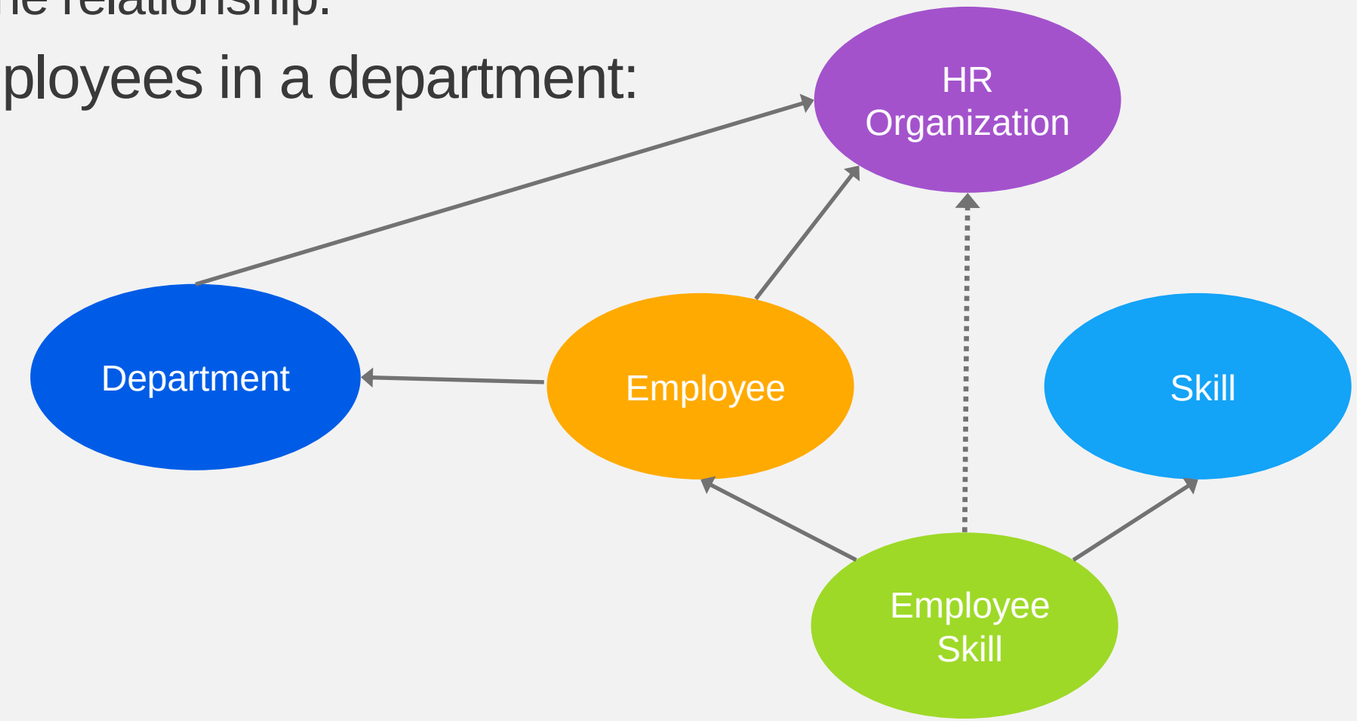
- First, we will define a new business object, Department.
  - Department needs the context of HROrganization to be valid.
  - Department will get its HROrganization context from Employee.
    - Employee is already in the context of HROrganization.



How to use relationships, continued

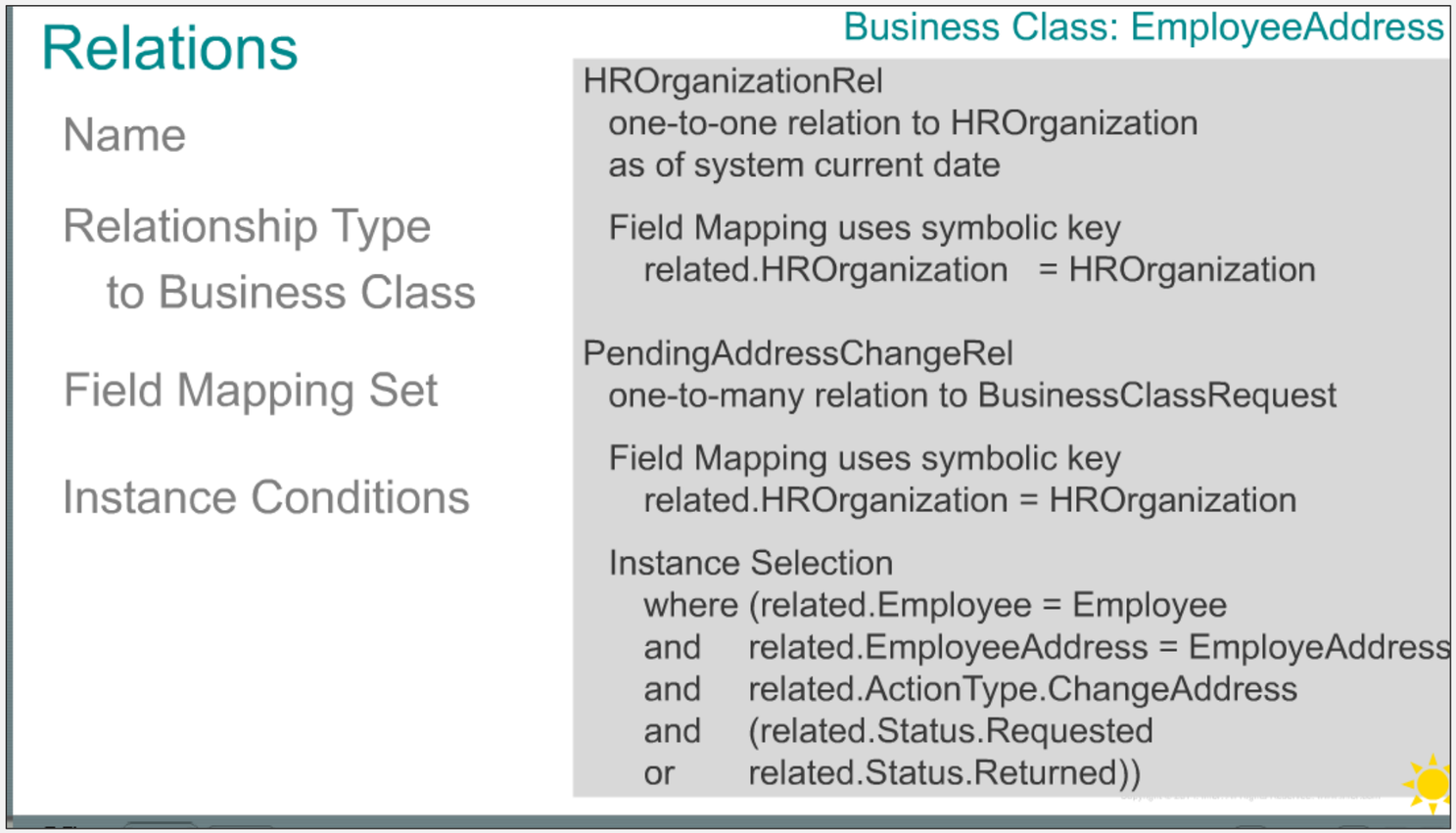
Names and sets

- If you want to know the supervisor of a particular employee's department:
  - Refer to Department.Supervisor.
  - There is an automatic one-to-one relationship.
- If you want to know all the employees in a department:
  - Refer to Employee.Set.



What a relation looks like in LPL

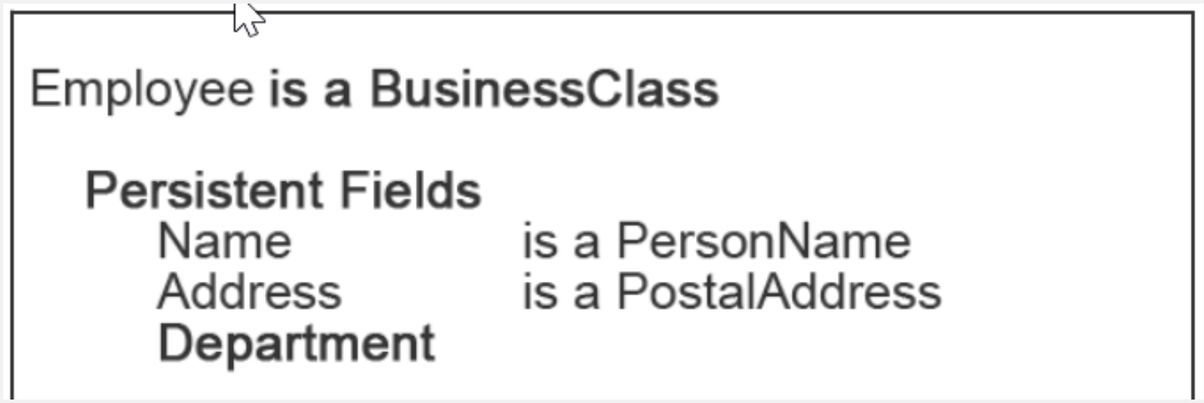
- Here is an example of the relationships for the EmployeeAddress business class.



How to use relationships, continued

Persistent fields

- Next, we will add Department as a persistent field on the Employee business class.
- Built-in relationships allow you to:
  - Navigate to that information.
  - Easily update these objects and understand what they are.



Check your understanding

Check your understanding

In terms of computer science, what is ontology?

Ontology is an organizing framework for understanding entities and relationships within a domain.

Check your understanding

Which two of the following statements about the Landmark ontology model are true?

a) It defines the order in which things must exist to have meaning.
b) It creates artificial relationships between objects.
c) Once created, the structure is permanent.
d) It is based on two primary objects: business classes and key fields.

Check your understanding

What does context refer to in terms of object relationships?

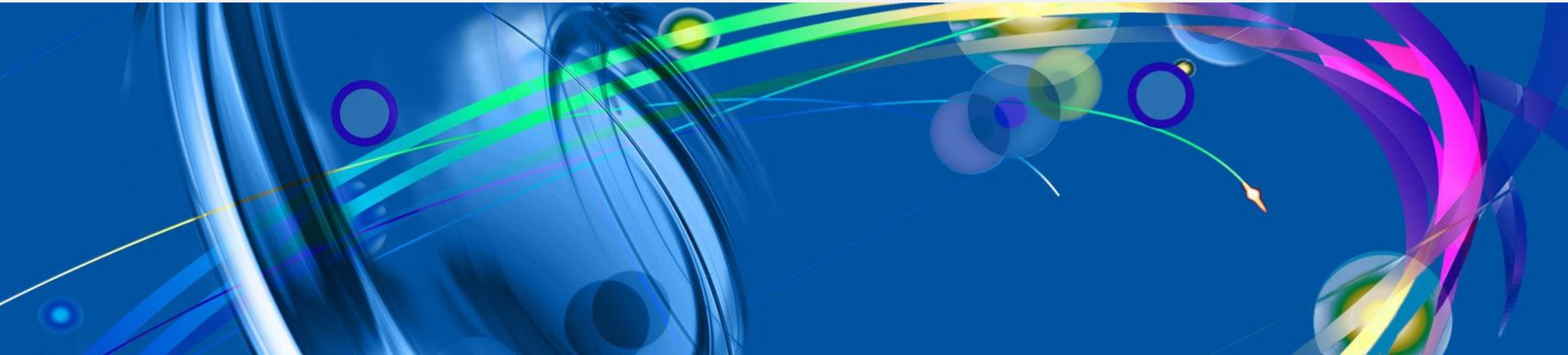
Context defines the objects that afford the existence of another object. For example, you must have a company to be able to define an employee.

Check your understanding

What method is used to navigate built-in one-to-one relationships?

a) Affordance
b) One-to-many
c) Dot notation
d) Sets

Lesson 3: Configuring business classes



Learning objectives

After completing this lesson, you will be able to configure a business class. In this lesson, you will:

- Describe a delivered business class.
- Discuss considerations for configuring a business class.
- Identify types of configurations that can be made to a business class.
- List the steps for configuring a business class.

Understanding business classes

- A business class is a table defined to store and persist data as well as actions, states, and security.
- A set of predefined business classes is delivered with Landmark.
- You can tailor a delivered business class by adding a configuration.
- Configuration questions:
  - What results do I want?
  - Does this change adapt how the application works, or does it modify the look and feel of application screens?
  - Which business class should I configure?
  - Do I have appropriate access?
  - How do I implement and test the change?
- If none of the delivered business classes can easily be configured to your specific business need, you can create a user business class.

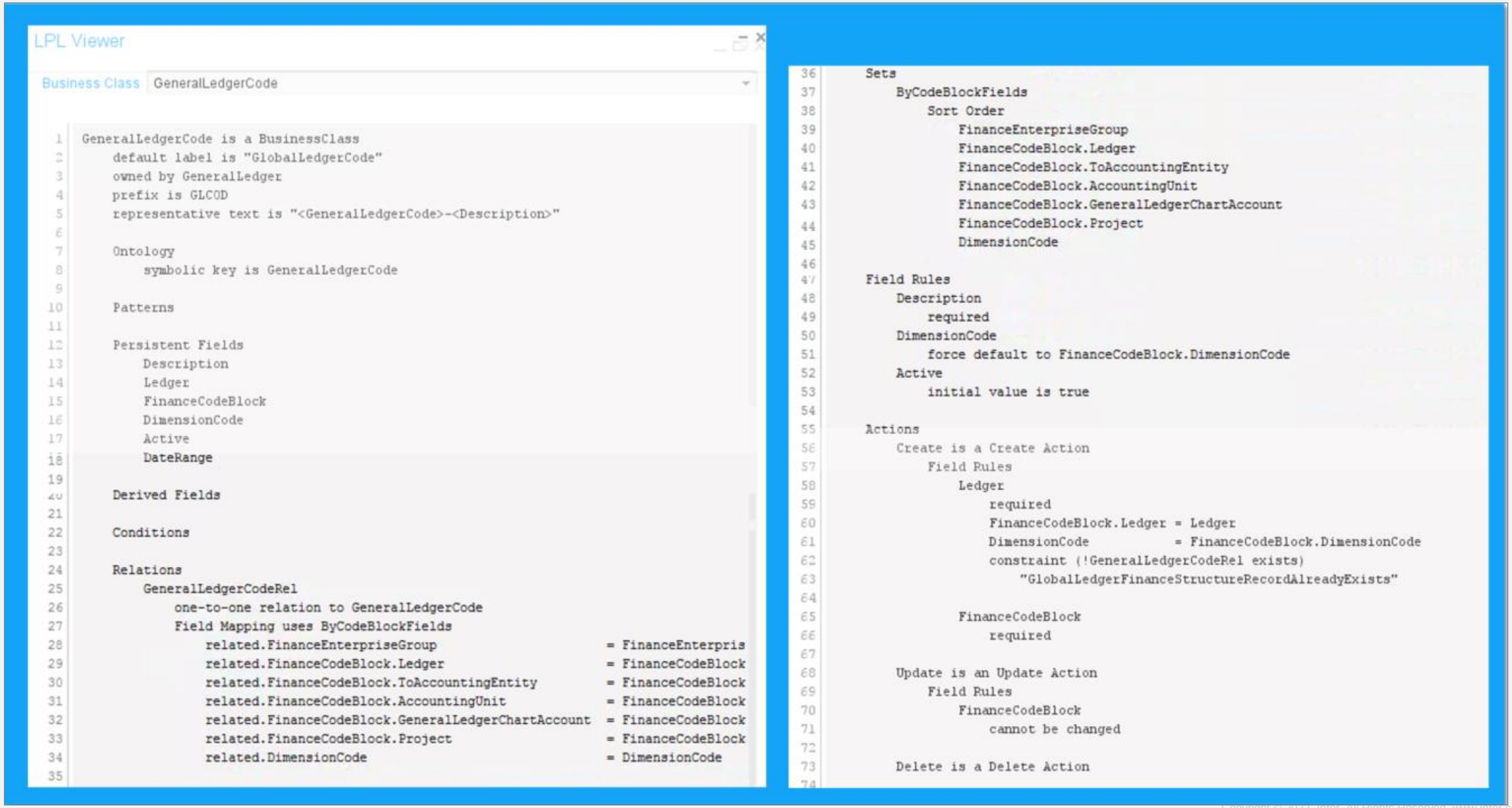
What can be configured?

- When configuring applications, the two types of configuration categories are:
  - Adapting applications to support organizational processes
  - Modifying the look and feel of application screens
- Both types are accomplished by configuring components of a business class:
  - Lists
  - Forms
  - Fields, buttons, check boxes and conditions
  - Actions
  - Action requests
  - Relations

What a business class look like in LPL

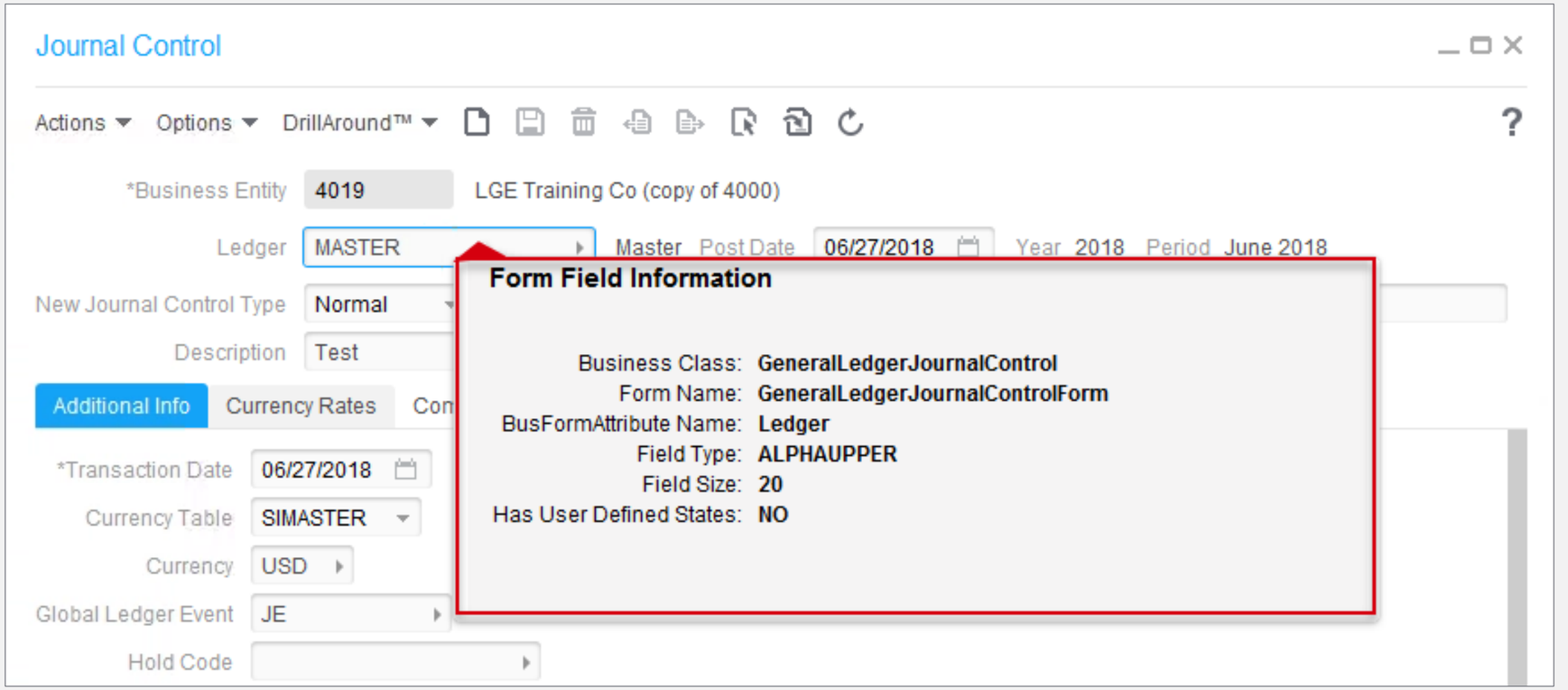
- Delivered business classes share a common LPL structure.
- LPL blocks:
  - Name
  - Ontology
  - Patterns
  - Persistent fields
  - Derived fields
  - Conditions
  - Relations
  - Sets
  - Field rules
  - Actions
- Even if a business class does not contain a component, the headings are typically still used in the LPL and left blank
  - Available if configurations are later made.

What a business class looks like in LPL, continued

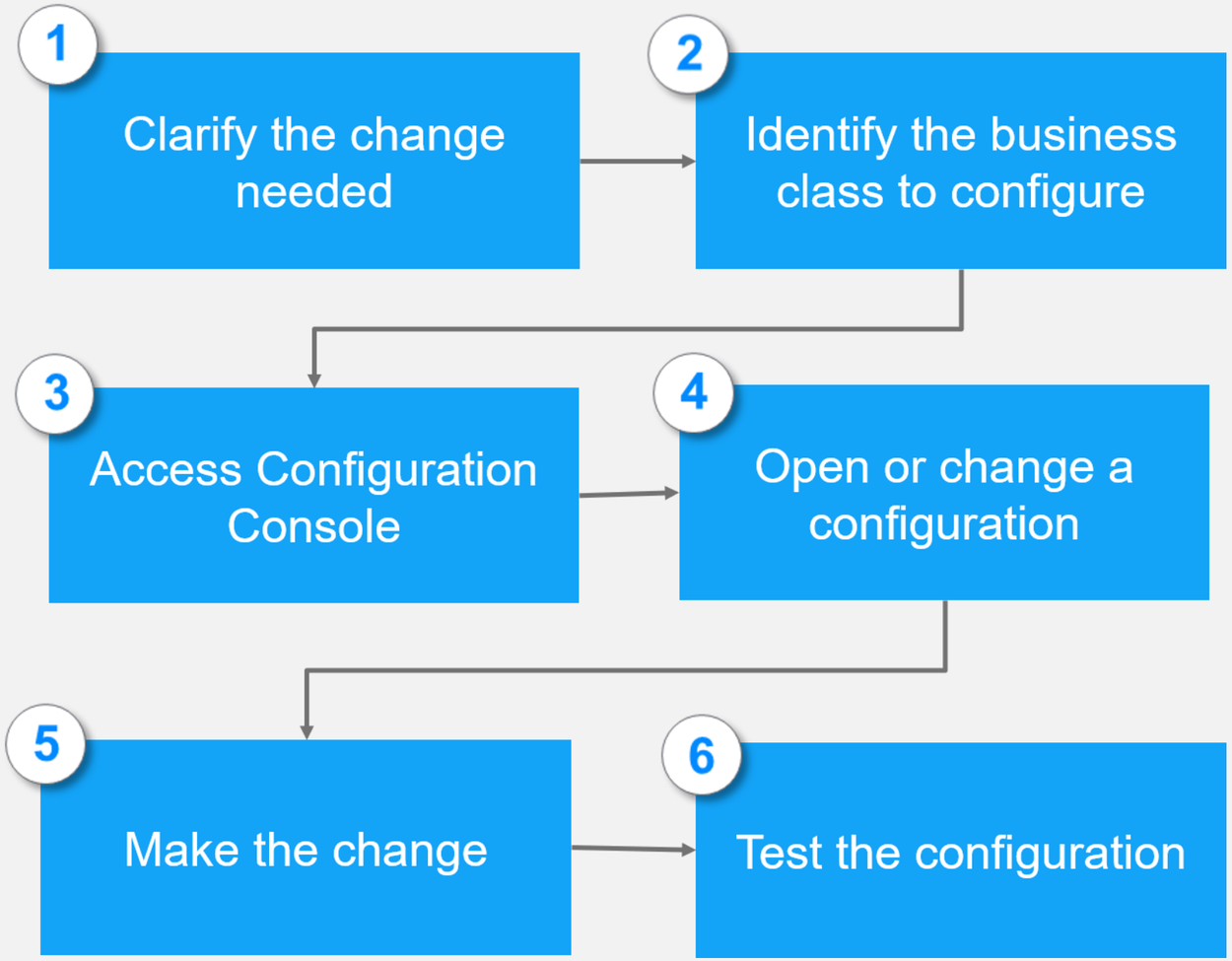


Identifying a business class

- Hover over an element such as a header, or place your curser in a field
- Press Ctrl + Shift + left-mouse click
- A Form Field Information window opens



Steps for configuring a business class



Background for Exercise 3.1

Your company has requested a modification to a delivered application. Before saving a posting record for FinanceDimension1, they want to ensure that an address code is entered. Summary records can be saved without an address code.

You know that there is an AddressCode field on the FinanceDimension1 business class, so you will configure this field to be required when saving posting records.

Exercise 3.1: Configure a delivered business class

- In this exercise, you will you will perform two simple configurations on a delivered business class. You will configure the Create and Update actions on the FinanceDimension1 business class. Then you will test the configuration on the Update action
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Check your understanding

Check your understanding

What is a delivered business class?

Business classes contain the basic definitions for what data is in an application, how it is organized, and how it is processed.

A delivered business class is one that Landmark provides to cover common business needs. It is ready to use out of the box or can be configured using Configuration Console.

Check your understanding

What questions should you consider before configuring a business class?

- What results do I want?
- Does this change adapt how the application works, or does it modify the look and feel of application screens?
- Which business class should I configure?
- Do I have appropriate access?
- How do I implement and test the change?

Check your understanding

How can you find out the name of a business class associated with a field?

a) Right-click in the field
b) Right-click any white space
c) Click in the field and press Ctrl + Shift + left-click
d) Click in the field and press Ctrl + F

Check your understanding

What are the two types of application configurations that use business classes?

a) Adapt applications to support organizational processes
b) Limit access to applications
c) Modify the look and feel of application screens
d) Import from and export to other data areas

Check your understanding

What is the final step when configuring a business class?

Test the configuration

Lesson 4: User business classes

Learning objectives

After completing this lesson, you will be able to configure a user business class. In this lesson, you will:

- Explain the difference between a user business class and a delivered business class.
- Describe naming conventions for user business classes.
- List the steps for creating a user business class.

About user business classes

- Created from scratch
- Has basic functions built in (create, update, delete) and you can add additional actions as needed
- Primary differences between user and delivered business classes:

User business class	Delivered business class
• Created by a user • Limited on what you can configure • If you edited the LPL code, you must use LPL to code any additional changes.	• Created by Infor • No limit on what you can configure • You can edit in LPL and later edit with dialog boxes.

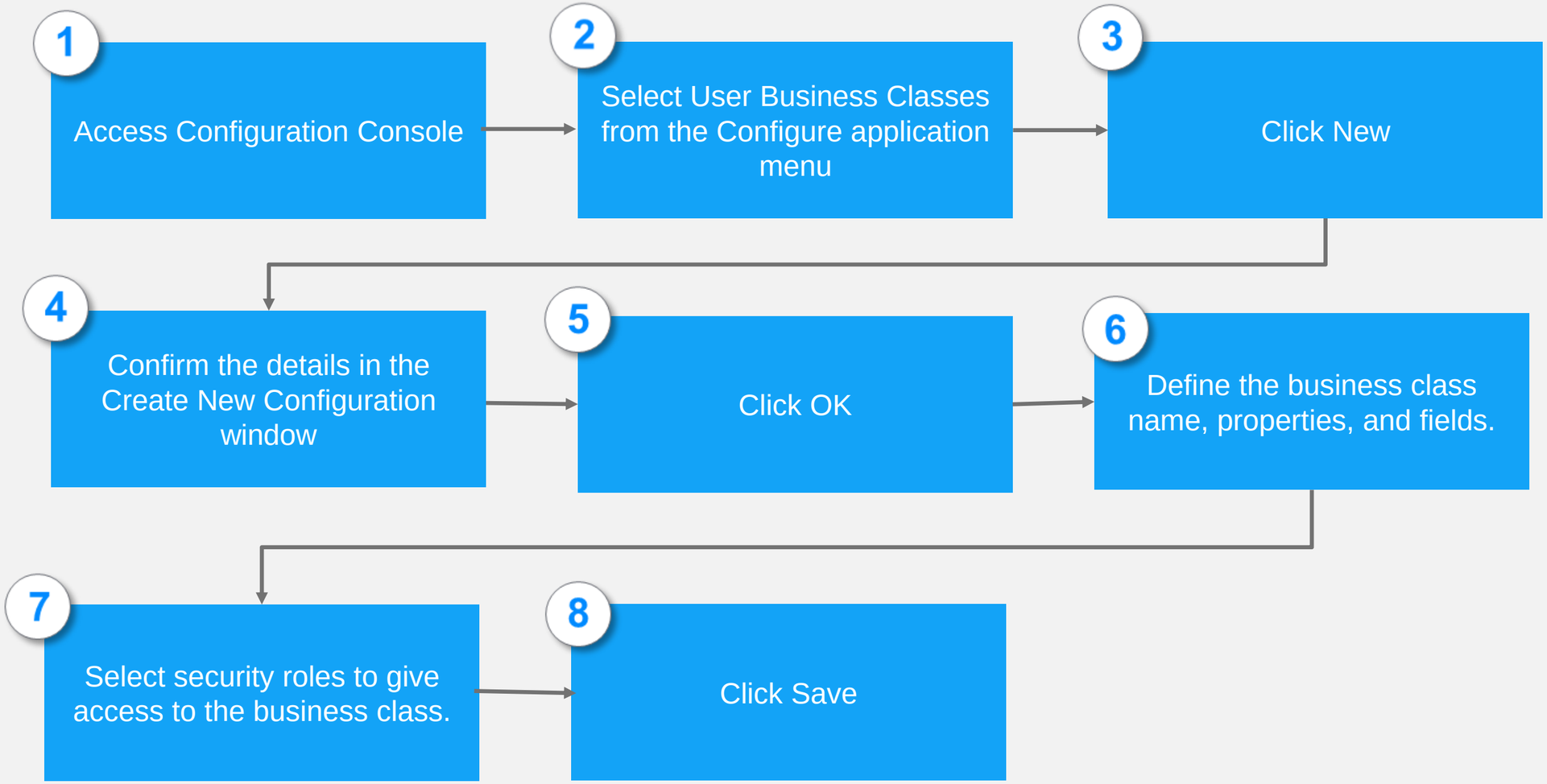
Naming conventions

- General naming conventions used for delivered Landmark business classes
  - Use a word or phrase that clearly describes the purpose/scope of the business class
  - Start with an uppercase letter
  - Contain no spaces
  - Each word or initial in a multiple word phrase is capitalized (i.e., CamelCase)
- When naming user business classes or other new components such as a new user field or form:
  - Follow the general naming conventions.
  - Add a unique organizational prefix to the name.
  - This makes the components easy to identify as being user created.

ABC Company tracks vendors in a non-typical way and decides to create a new user business class for vendors rather than configuring the delivered Vendor business class.

What is an appropriate name for this new business class and why?

Steps for configuring a user business class



Background for Exercise 4.1

Company XYZ wants to track business partners in a custom manner, and needs a special business class for this purpose.

The company has determined it is not feasible to modify any of the existing business classes that were delivered with Infor Landmark, and instead wants you to create a user business class from scratch.

Exercise 4.1: Create a user business class

- In this exercise, you will create a user business class for business partners using Configuration Console, and then review the LPL code generated
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Check your understanding

Check your understanding

What are some differences between user business classes and delivered business classes?

A user business class is created by a user from scratch. There are limits on what you can configure, and if you edit the LPL code at any time, you must always edit the LPL.

A delivered business class is created by Infor and delivered with the Landmark system. There is no limit to what you can configure, and you can edit in LPL or dialog boxes.

Check your understanding

List naming conventions for a user business class.

- Includes a unique organizational prefix that identifies it as a user business class
- Follows other conventions of delivered business classes
  - Uses a word or phrase that clearly describes the purpose/scope
  - Starts with an uppercase letter
  - Contains no spaces
  - Each word or initial in a multiple word phrase is capitalized

Check your understanding

What steps for creating a user business class differ from configuring an existing business class?

Create and name the business class

Assign security roles for access

You also have to define properties and fields, though this is similar to what you do for configuring an existing business class.

Lesson 5: User fields

Learning objectives

After completing this lesson, you will be able to create user fields. In this lesson, you will:

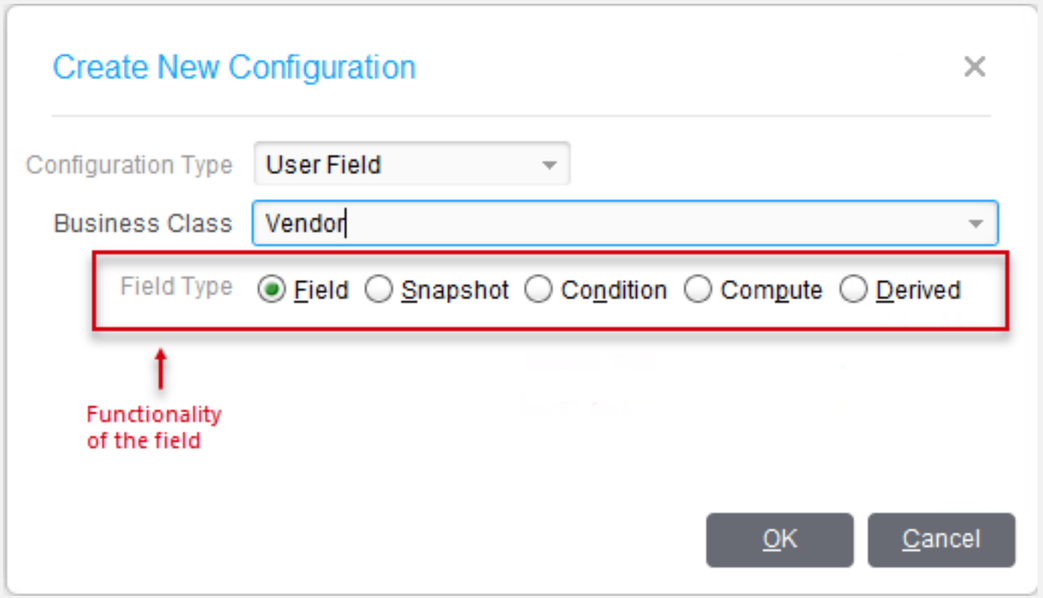
- List the types of fields.
- Compare persistent and transient fields.
- Explain how to add states to a user field.
- Create a derived field.
- Describe how to add fields to a business class.

About fields

- Fields can be placed on forms and lists.
- They are a way for users to view, enter, or interact with data.
- A set of common field objects are delivered with the Landmark system.
  - You can use these field objects, but you cannot configure them.
  - This means you cannot change the behavior of a delivered Landmark field object such as PostalAddress.
- If the delivered field objects do not meet your needs, you can create new fields from scratch.

Creating a new field

- Select Business Classes in the Configure application menu.
  - If a business class configuration already exists for the business class you want to add a field to, expand the business class, select User Fields, and click the New button to open the Create New Configuration form.
  - If a business class configuration does not exist, select the New button at the top of the left pane and then select User Field as the configuration type in the dialog box that appears.
- Notice that you must select the type of field to create.

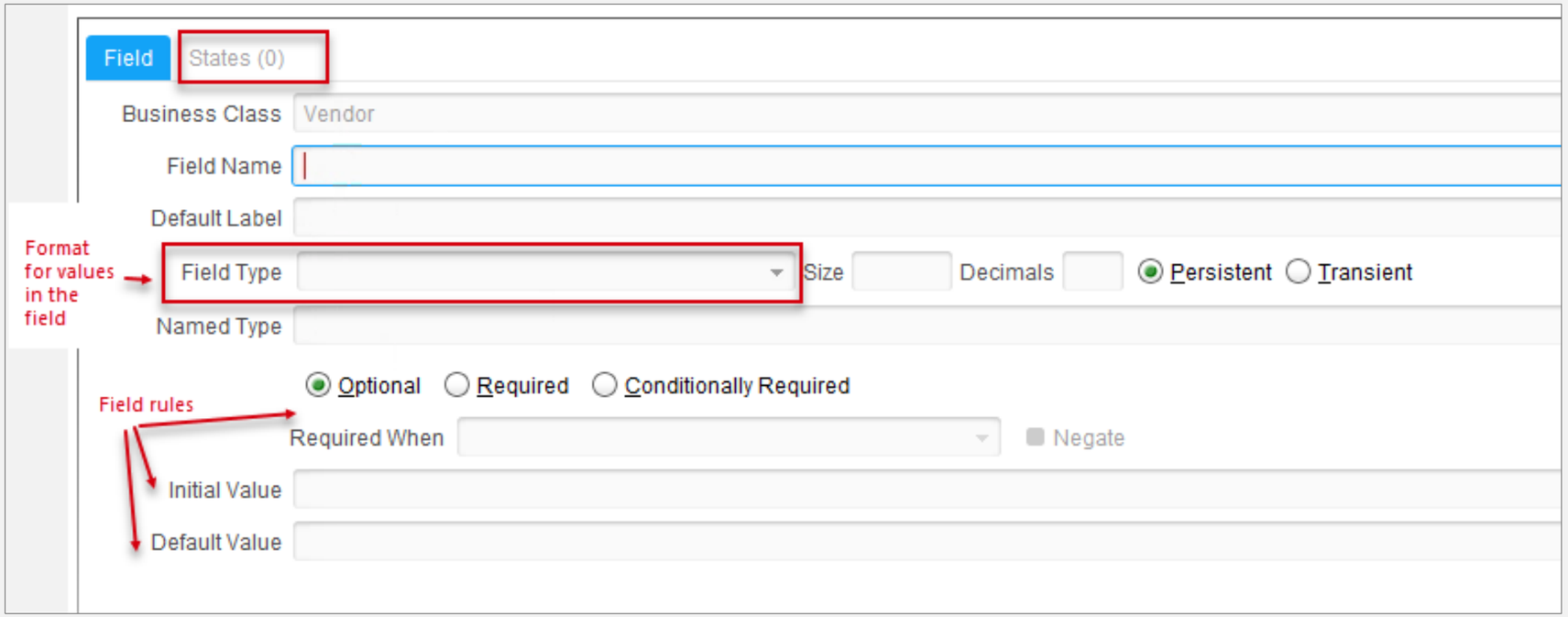


Types of fields

Field type	User interaction	Description
Field	Users enter data into fields.	User fields enable you to add a new field to a form or list where you define the name, type, and length of the field. Users can then enter data in the field and data will be saved.
Snapshot	Users will not type or specify anything your new field, the value in the field comes from the snapshot source.	Use a snapshot to capture values from another field, such as capturing an invoice number in a field on a general ledger transaction. You can choose from existing business class fields to create a user field of the same type and size that copies the value from the original field to your new field if the data source is in context. In other words, the value of the snapshot source displays in the new snapshot field.
Condition	Users see a display-only check box that shows the status of the condition or they see a selectable check box to control a condition, depending the condition.	You can add conditions to a business class definition. This can display the condition or control the behavior of other user interface components, such as controlling whether a field is visible or not.
Compute	Users see the computed value in this display-only field.	A compute field is a statement that uses values from existing fields, static values, or mathematical operations
Derived	Users see the derived value in this display-only field.	Derived fields are written with LPL code by using existing values, operations, or scripts

Defining a user field

- Fields are defined on the Configure form that displays in the right panel of the Configuration Console.



Defining a user field, continued

Type

- You must provide a field name and select a field type or named type.
- Field type (on this form) defines the format for values in a field. Here are a few of the common static field types:
  - Alpha – Allows upper case and lower case alphanumeric characters
  - AlphaUpper – Forces only uppercase alphanumeric characters to avoid confusion with cases
  - Boolean – 0 = false, 1 = true; to check if the Boolean is true, we check that the field has a value
  - Numeric – Allow numbers only
- Named type adds a business class as a user field
  - When you select a named type field, the drop-down list of values a user sees are records in the business class.

Defining a user field, continued

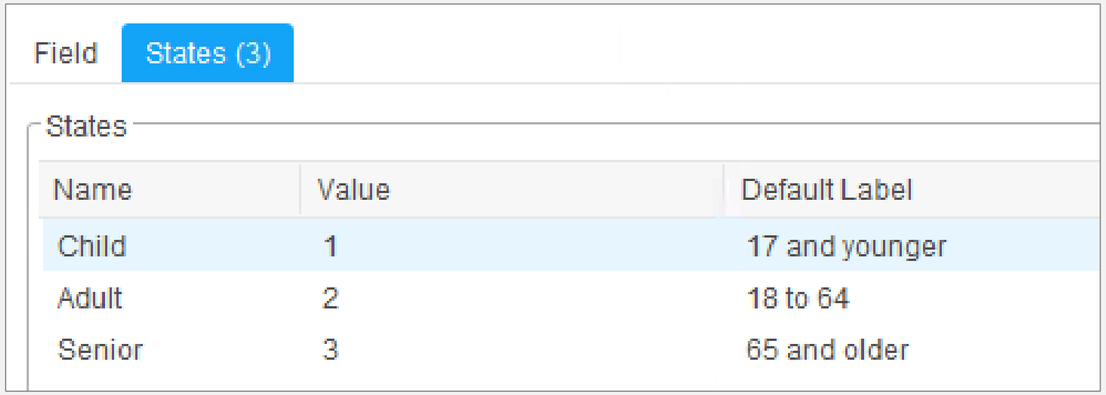
Field rules

- Field rules and conditions are optional
- Here are a few common field rules:
  - Initial value – Value of the field that appears in the form automatically pre-populates with data form another business class or a constant value
  - Default value – If you enter nothing in the field, the field will take the default value which can be from another business class or a constant.

Defining a user field, continued

States

- States are used for fields where it is appropriate to provide a drop-down list of possible values.
- Enter states on the States tab when defining a user field.



- Not all field types support the use of user-defined states.
  - For example, a Boolean fields field has a built-in states so you cannot add custom states to a Boolean field.

Persistent and transient fields

- User fields can have differing functionality, but they fall into one of these categories:
  - **Persistent** – Values are saved in the database (this is the default)
  - **Transient** – Values are not saved in the database
- Persistent fields are used to gather data and can also be used to associate business classes.
  - For example, when a department field is added to the Employee business class, it creates a one-to-many relationship between Department and Employee.
- Transient fields are similar to parameters.
  - You can act on data entered in your logic, and use the values in calculations.

Exercise 5.1: Configure persistent and transient fields

- In this exercise, you will add five persistent user fields and one transient user field to the FinanceDimension1 business class
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Conditional fields

- You can add conditions to a business class definition.
- These conditions can be:
  - Display-only check boxes that show the status of the condition.
  - Conditions that control the behavior of other user interface components, such as controlling whether a field is visible or not.

Exercise 5.2: Configure a conditional field

- In this exercise, you will add a conditional user field to the FinanceDimension1 business class
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Compute fields

- Created through a statement that uses values from existing fields, static values, and/or mathematical operations.
- Display only
- Several tools or shortcuts are available to you to create a compute statement.
- After you configure a compute field it will be available to be added to a form configuration.
- You can refer to a compute field when you define a derived field.
- Example

  ElapsedTime field

  (current time - StartTime)

Derived fields

- Variables that do not exist in the data, but are created from existing fields
- Display only – they are not saved in the data
- May include using mathematic operations with existing fields
- Defined using LPL syntax and always include a “return” statement, describing what value to display
- Example:

```
SalesTax is a DerivedField
    type is Numeric size 5
    return (Cost * .05)
```

Exercise 5.3: Configure a derived field

- In this exercise, you will add a derived user field to the Account business class. You will then add these new user fields to the FinanceDimension1 business class:
  - An instance count field
  - A math compute field
  - A derived concatenated field
  - A derived date math field
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Adding user fields to a business class

- A standard form may not contain all of the fields needed for your business.
- Using the Configuration Console, you can add fields to a business class configuration.
- After the fields are defined, you can then add them to a business class.

Exercise 5.4: Add fields to a business class

- In this exercise, you will add a string field and a named persistent field to the XYZBusinessPartner user business class using LPL
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Exercise 5.5: Add a field to a business class that is a key field for a user business class

- In this exercise, you will add a persistent field to the FinanceDimension1 business class, which is a keyfield for the XYZBusinessPartner user business class
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Check your understanding

Check your understanding

List the five types of user fields.

- Field
- Snapshot
- Condition
- Compute
- Derived

Check your understanding

What is the difference between a persistent field and a transient field?

- A persistent field is saved in the database. A transient field is not saved.

Check your understanding

If you add states to a field, how will this affect the functionality of the field?

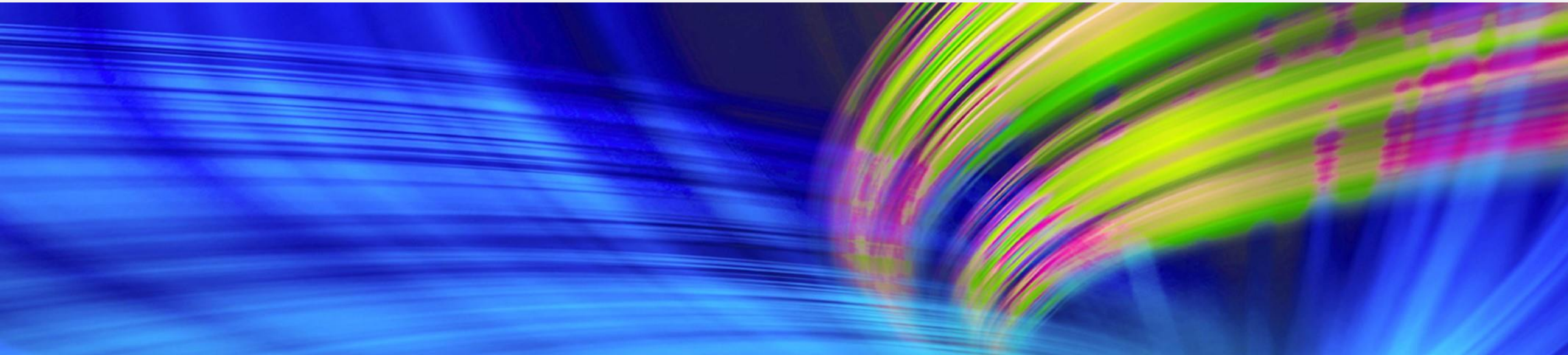
States are visible in a drop-down list for the field.

Check your understanding

What is a derived field?

A derived field is one that displays a value derived from other data and may include mathematical operations. These fields are designed using LPL syntax and typically include a return statement.

Lesson 6: Configuring user interface objects



Learning objectives

After completing this lesson, you will be able to configure common user interface objects. In this lesson, you will:

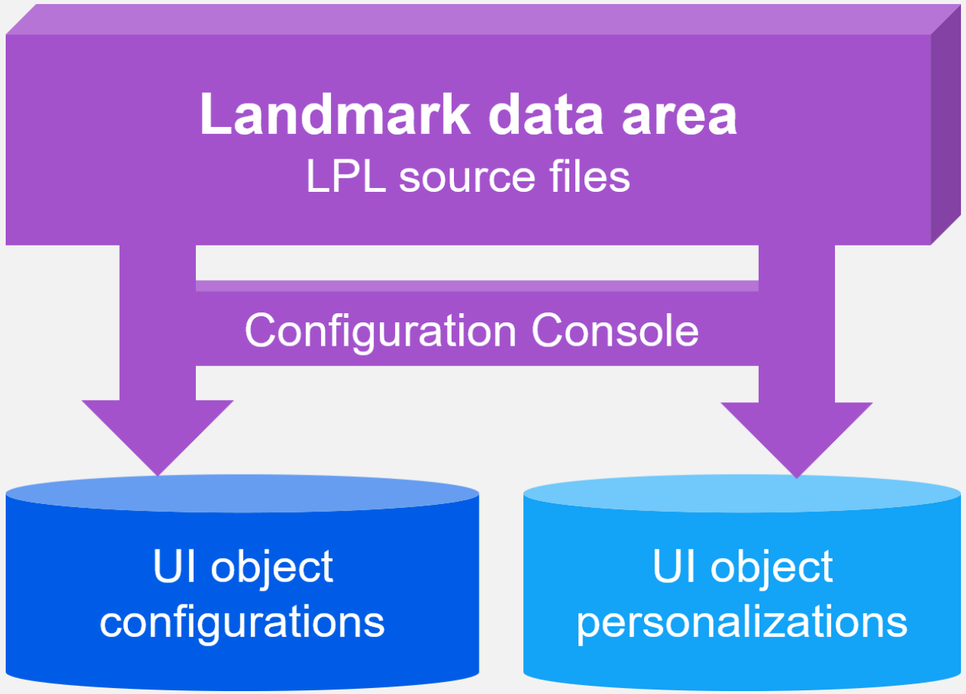
- Identify common user interface objects.
- Compare configuration with personalization.
- Configure a form.
- Describe how LPL controls the display and functionality.
- Create a new menu item.
- Configure a list.
- Configure a composite form.

Configurable user interface (UI) objects

- Applications use several types of user interface objects that enable users to interact with the application.
- Application configurations can modify the interface to better reflect how the organization does business.
- Types of UI objects you can configure:
  - Pages
  - Forms
  - Lists
  - Menus
  - Composite forms

LPL source files

- To configure a UI object, you need to modify the LPL code for the object.
- LPL source files can be accessed via Configuration Console and then saved out as configurations or personalizations in a separate database.
- The LPL source files are not affected and remain unchanged.



Configuration vs personalization

Configurations

- Changes made to a delivered object such as a list, form, or menu that makes it easier to work with
- Affects all users of the application
- Require special security to create

Configuration vs personalization, continued

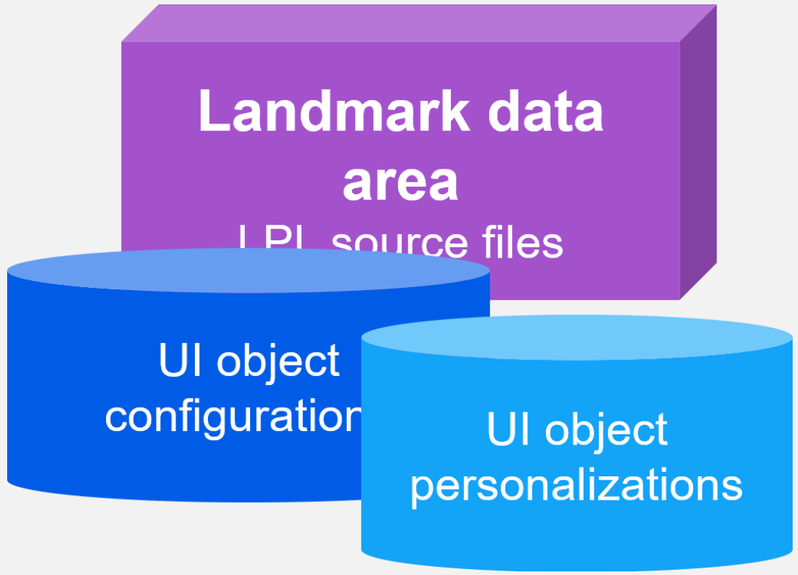
Personalizations

- Changes made to a delivered object such as a list, form, or menu that makes it easier to work with
- Affects only the end user who made the personalization
  - If two different users both personalize the same object, their experience using the object will differ.
- Do not require special security
  - Not all personalizations are available to all users because your system administrator controls access to personalizations.
- Example
  - Jason and Mary both use the inventory list. Mary added a personalization to reorder the columns so that Quantity appears in the first column. When Jason and Mary each open the inventory list, the data that makes up the list is the same, but their views will differ.

Configuration vs personalization, continued

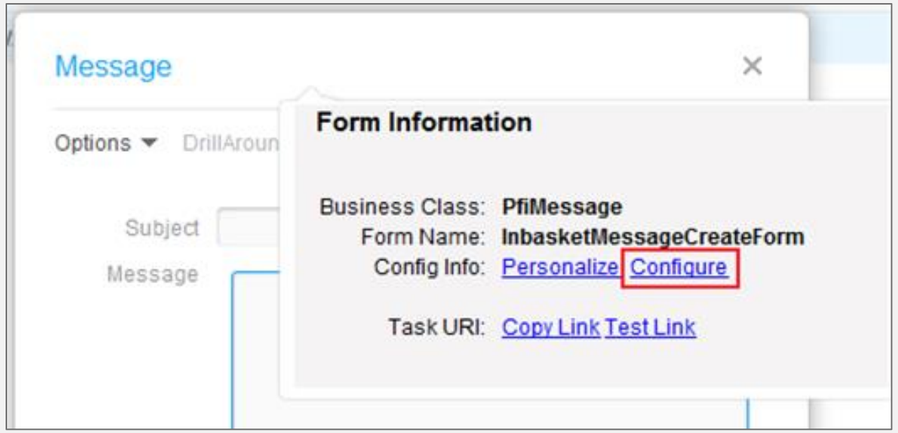
How configurations and personalizations are applied

- Configurations and personalizations are dynamic and take place in real time.
- Here is what happens when a module is launched:
  - First, any configurations created for objects within the module are brought into memory as an overlay of the LPL source files. The configurations are executed in place of the source files.
  - Then, for the user launching the application, any personalizations created by the user execute, overlaying and superseding any configurations.



Starting points for configuring UI objects

- There are two ways to access Configuration Console to configure a UI object:
- **Method 1: Start from Configuration Console**:
  - Select Start > Configure > Application.
  - Select the item in the Application menu for the type of object you want to configure.
  - Click New.
- **Method 2: Start directly from the object you want to configure**:
  - Open the object, such as a page or form.
  - Press Ctrl + Shift + left click in the header.
  - Click the Configure link.



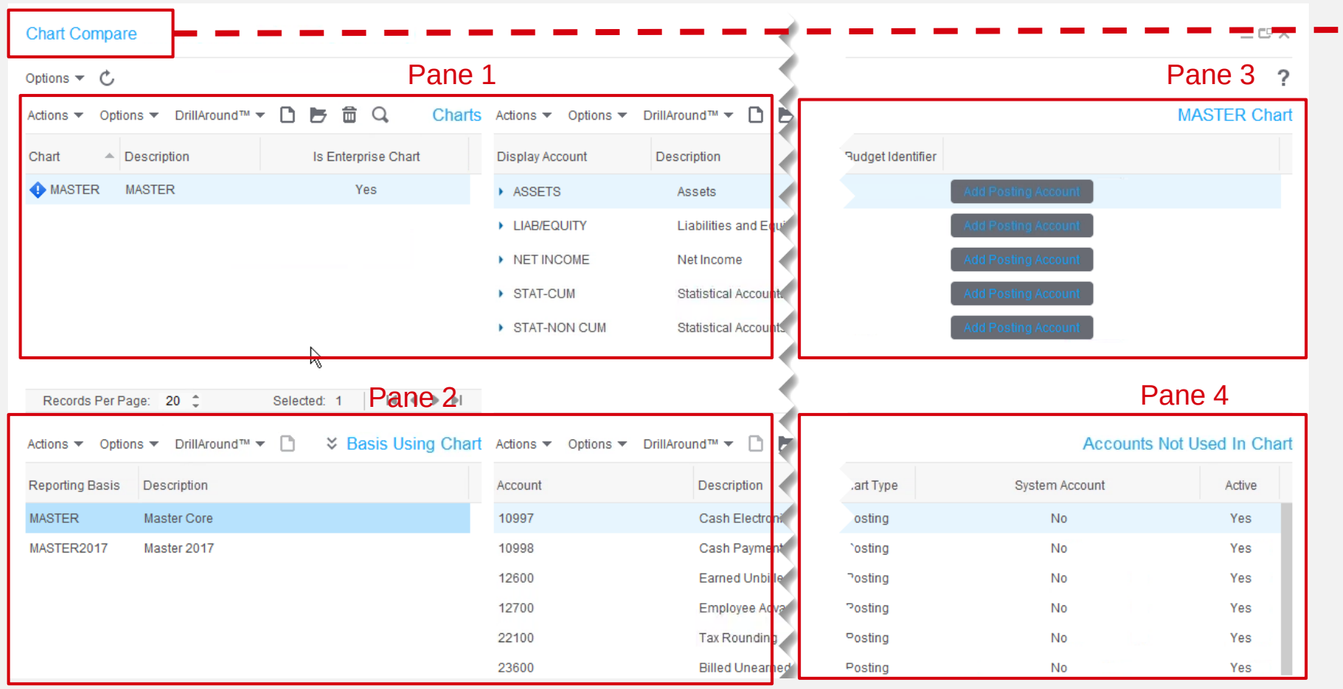
Configuring pages

About pages

- Allow you to display several UI objects in one view
- Contain a set of tabs or panels that enable you to access lists and forms
  - The business class for each panel must be specified in the LPL code.
  - Each panel can contain a menu, a form, a list, or a URL.
- Can include objects from more than one business class, whether they are related or not.

Configuring pages, continued

Example page




```
ChartAccountCompare is a Page
    title is "ChartCompare"

    ChartAccountCompare is a MultiPanePanel
        pane 1
            business class is ReportingChart
                list is ReportingChartsListForComparePage
        pane 2
            business class is pane 1.ReportingBasis set
                list is BasisUsingChart
        pane 3
            business class is pane 1.ChartAccountNode.ChartAccountChildrenRel
                list is ChartAccountList
        pane 4
            business class is pane 1.NotInReportingChartAccountRel
                list is UnusedAccountByChartList
```

Configuring pages, continued

Types of page configurations

- Create a new tab on the page
- Remove a tab
- Change the page design to include multiple columns
- Add a list to a new panel
- Modify a list

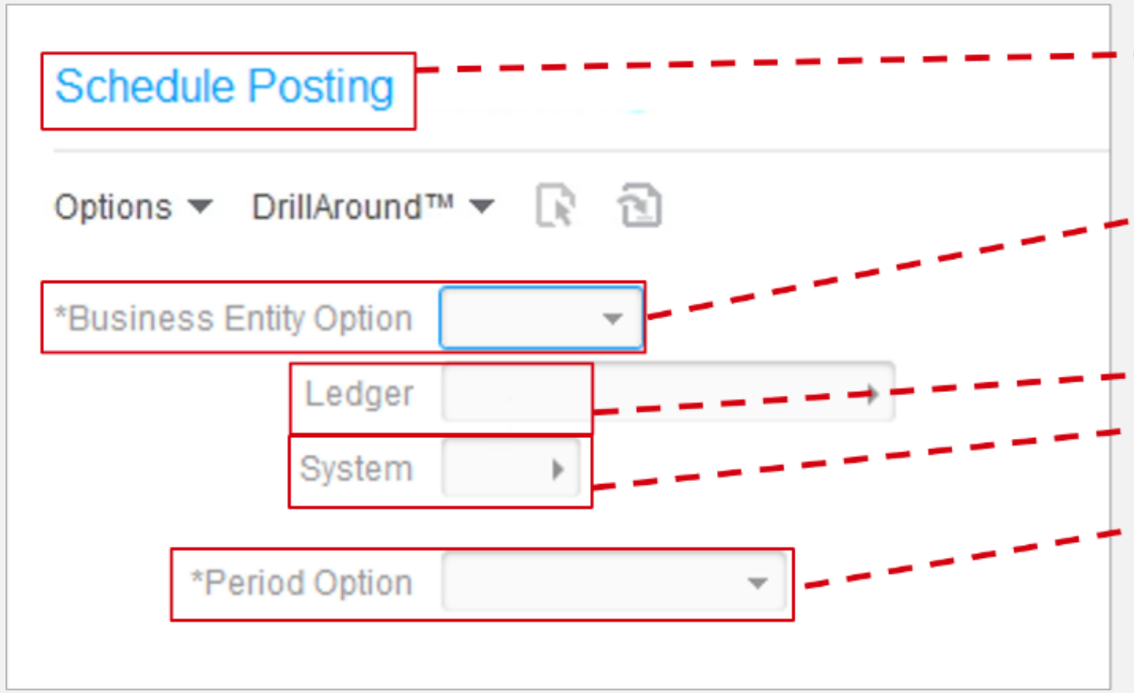
Configuring forms

About forms

- Allow you to interact with a business object (perform a business task) in an intuitive arrangement
- May include fields, buttons, check box controls and actions
- Can have a variety of associated actions
  - For example, a form may allow you to create or update data records, register a customer or candidate, or start an approval work flow.
- LPL code for a form specifies the layout and the fields for entering record data.

Configuring forms, continued

Example form




```
SchedulePosting is a Form
    is primary
    Layout
        single column
            AccountingEntityOption
                label is "<FinanceEnterpriseGroup.AccountingEntityLabel>Option"
            visible when (AccountingEntityOption.Single)
                AccountingEntity
                    label is "<FinanceEnterpriseGroup.AccountingEntityLabel>"
            visible when (AccountingEntityOption.Group)
                AccountingEntityGroup
                    label is "<FinanceEnterpriseGroup.AccountingEntityLabel>Group"
            Ledger
                select is WithoutCloseLedgerList
            System
        blank line
        single column
            PeriodOption
        visible when (PeriodOption.SpecifyPeriod)
            two column
                Period
                    select is ClosePeriods
                Period.Date
                    no label
                    display as text
```

Configuring forms, continued

Types of form configurations

- Change the form heading
- Change which actions display in the toolbar
- Make actions conditional
- Reorganize components within a section
- Change the form layout by adding new:
  - Sections, headers, or paragraphs
  - User fields
  - Blank lines, columns, or page breaks
  - Conditions

Exercise 6.1: Configure a form

- In this exercise, you will add the user fields you created with the XYZ prefix in previous exercises for the FinanceDimension1 business class to a delivered form, UserFieldsForm. This will display as a User Fields tab on records for the FinanceDimension1 business class.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

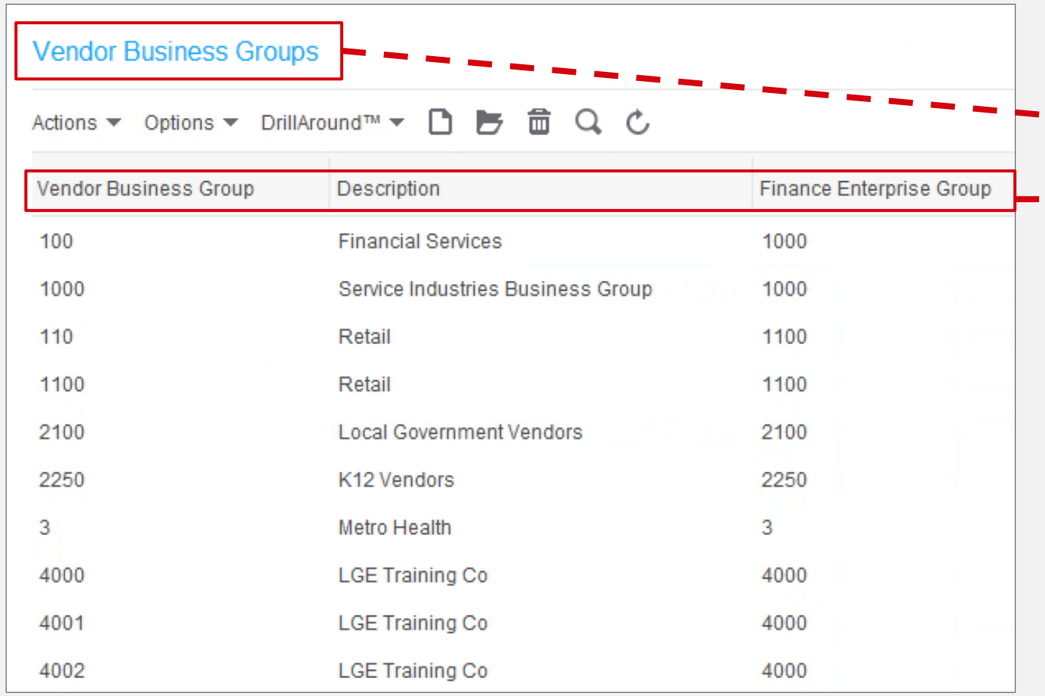
Configuring lists

About lists

- Simple presentation of records in column format
- Present data from one business class and its related classes
- You can select an item in the list and open a form to change the data or perform an action for that record.
- LPL code for a list specifies the data to be displayed in the list field columns
  - It also specifies the actions that are available and restricted for the list.

Configuring lists, continued

Example list




```
1  BusinessGroups is a List
2      is primary
3      title is "VendorBusinessGroups"
4      Display Fields
5          BusinessGroup
6          Description
7          FinanceEnterpriseGroup
8
```

Configuring lists, continued

Types of list configurations

- Change the list title (heading)
- Add, delete, or rearrange columns
- Add alerts
- Define a filter so the list presents only a subset of the available records
- Modify the actions that appear in the list toolbar.
- Define how values display (e.g., negative numbers, rounded figures)

Configuring lists, continued

Inline create (add row)

- By default, creating a new record from a list opens a separate Create form.
- You can improve usability for simple data entry by implementing InlineCreate syntax in your configured or user-defined lists.
  - Allows the user to create a record directly within the list
  - Adds a blank row in the list where the user can complete record entry
- If implemented, the only way to enter records is within the list.
- You can also include Allow Form Create syntax so the list allows records to be created both inline and with a form.

Exercise 6.2: Configure a list and create a new list

- In this exercise, you will configure the default list and form that is generated for the user-defined business class XYZBusinessPartner. Additionally, you will create a new list for that business class.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

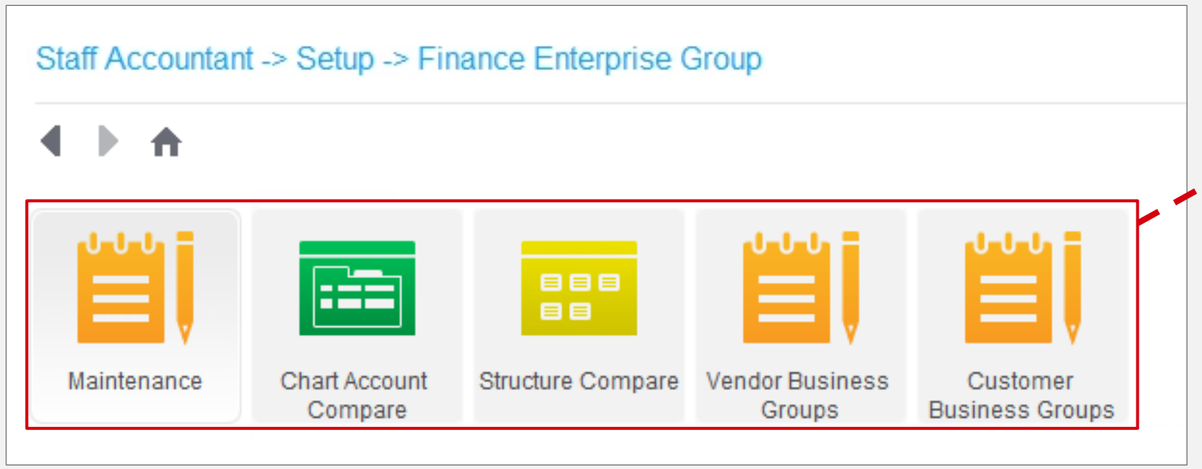
Configuring menus

About menus

- A menu consists of a set of menu items.
- A menu item can route the use to a page, a form, a composite form, a list, or another menu.
- You can add or remove items from the standard delivered menus in Infor Rich Client.
  - For example, you may find submenus that are highly specialized can be removed from the general roles.
- Changes to a menu will be reflected for all users who have access to the menu.

Configuring menus, continued

Example menu



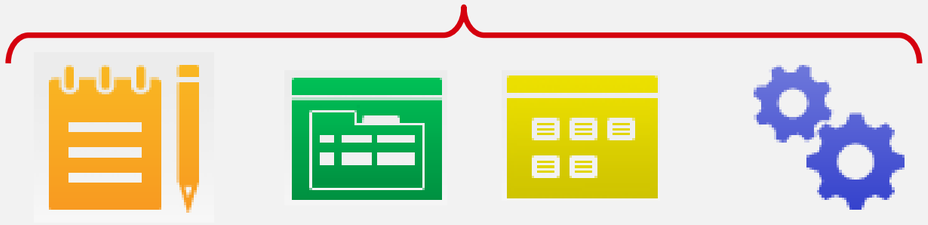

```
1  Setup is a Menu
2      Menu Items
3          Maintenance
4              list is FinanceEnterpriseGroup.FinanceEnterpriseGroupList
5          ChartAccountCompare
6              page is ChartAccountCompare
7          StructureCompare
8              menu is StructureCompare
9          VendorBusinessGroups
10             list is BusinessGroup.BusinessGroups
11         CustomerBusinessGroups
12             list is CustomerBusinessGroup.primary
```

Configuring menus, continued

Types of configurations

- Remove a menu item
- Add a menu item
- Change menu item title

Icons used on springboard menus in Infor Rich Client



List icon | Page icon | Menu icon | Form icon

Background for Exercise 6.3

Currently, the only way to access XYZBusinessPartner data is to use the search to find a link to the list. You will create a new menu item to access this list directly. Here is the menu you will configure:

Exercise 6.3: Configure a menu

- In this exercise, you will configure a menu so you can access the XYZBusinessPartner Department List from a menu item.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Configuring composite forms

About composite forms

- Composite forms, like pages, contain a set of tabs or panels that enable you to access lists and forms.
- A composite form is created within a business class, so the LPL codes do not need to specify the business class.
- Composite forms do not allow the more extensive field and layout changes that regular business class forms allow.
  - However, you can create separate configurations for the forms contained within a composite form.

Configuring composite forms, continued

Types of configurations for composite forms

- Change the form heading
- Change the look and feel of the form in one of these ways:
  - Show panel controls on left
  - Show panel navigation
  - Show steps
- Change the form layout by adding, moving, or editing panels.
- Modify what actions appear in the toolbar
- Add a menu item

Exercise 6.4: Configure a composite form

- In this exercise, you will configure a composite form on the user-defined business class XYZBusinessPartner.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Check your understanding

Check your understanding

If you define a configuration that does not seem to take effect, what is the first thing you should check?

Check that you do not have a personalization that is masking your change.

Check your understanding

Match each of the following descriptions with the interface object it defines. The possible interface objects are: **Menu**, **Form**, **Composite form**, **Page**, and **List**.

Description	Interface object
Create, view, and edit a record	Form
View multiple records; select a record	List
Tabs access lists and forms for different business classes	Page
Tabs access lists and forms within a business class	Composite form
Icons access pages, forms, composite forms, and lists	Menu

Check your understanding

What does the LPL code for a menu specify?

It specifies the name for each menu item and the user interface object that each menu item accesses.

Check your understanding

What term in LPL code do you use to specify a tab?

a) Form
b) Page
c) Header
d) Panel

Check your understanding

Within a composite form, what can a tab access?

Other forms and lists

Check your understanding

What is specified in the LPL code for actions that pertain to lists?

LPL code specifies actions that can be performed on list items as well as a set of actions restricted from use.

Lesson 7: Configuring actions

Learning objectives

After completing this lesson, you will be able to configure an action. In this lesson, you will:

- Identify the set of delivered actions.
- Describe methods for configuring an action.
- Configure an existing action.
- Describe entrance and exit rules.

Understanding actions

- Business classes can have actions and action requests
  - For example, the Employee business class may include actions such as hire resource, transfer, promote, change pay rate, add work assignment, and terminate.
- There is a set of actions delivered with Landmark applications that include the following:
  - Create actions - create new records.
  - Update actions - update existing records.
  - Delete actions – flag a record as deleted; it can no longer be acted upon but the record remains and can be reported on as a deleted record.
  - Instance actions - use rules, unique to the action, for a single record.
  - Set actions - use rules, unique to the action, for a set of records.
  - Purge actions - remove records from the database.
- You can configure actions or create new actions from scratch

Methods for configuring actions and action requests

- **Configure an existing action or action request**
  - You can modify an existing action in the following ways:
    - Modify if it appears in the toolbar of a form
    - Hide the action always or only when a condition is met
    - Override an action setting, such as whether a comment or effective date is required
  - Configuring does not modify the underlying business logic.
- **Create an extension to an existing action or action request**
  - User extensions customize an existing action and create specific forms for the action.
  - They continue to use the business logic of the action they extend but the form can be unique for each user action.
  - For example, you can add a new Create action that extends an existing Create action and then create or copy a form to a new User Form so that the new Create action can bring up the new form.

Methods for configuring actions and action requests, continued

- **Create a new action**
  - You can create a new action "from scratch" for a business class for action types that are not action requests.
- **Create an action request**
  - Create a new action request from scratch.
  - Please note that it is most useful to create an action request, only if action requests are not already defined for a business class.
  - If action requests are already defined, there is often important business logic involved in the processing of the action request that you will not be able to duplicate if you create your own action request. In those cases it is preferable to create an action extension instead.

Business logic

- Business classes, actions, and other files built with LPL encapsulate all application business logic.
- You can add additional business logic to actions by adding other elements to the LPL code for an action such as:
  - Rules
  - Looping constructs
    - While
    - For each
  - Subscripts

Exercise 7.1: Configure an existing action by adding business logic

- In this exercise, you will add a configuration to an action on FinanceDimension1. You will add a Boolean user field (a transient field) to a form. You will also configure the Create action to have a second rule that says if the Boolean transient field is true, turn active off.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Entrance and exit rules

- You can define your own entrance and exit rules for delivered LPL actions.
- Entrance rules execute before any existing rules.
- Exit rules execute after all other existing rules.
- Entrance and exit rules must follow LPL syntax.
- They can access additional services outside of Landmark applications.
- They cannot be created for business classes that are stored in the Landmark environment (GEN).

Background for Exercise 7.2

Company XYZ wants to make sure that no vendor can have the same Tax ID as another vendor.

You will configure the Update action on the Vendor business class with a constraint that ensures no other vendor can have the same tax ID number as the selected vendor.

Note: If you were implementing this for your organization, you would want to do it on other actions such as the Create action as well. For this course, we will only configure the Update action.

Exercise 7.2: Configure a user relation and use it to configure an action

- In this exercise, you will first create a user relation on the Vendor business class, and then use that relation to configure an action.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Background for Exercise 7.3

Your organization wants to make sure that only numeric characters are used for subaccounts.

To do this, you can use a feature in Landmark to see if a field matches a regular expression. A regular expression allows you to define which patterns in a string you want to match against.

Exercise 7.3: Configure an action that prevents invalid characters

- In this exercise, you will configure the Create action on the GeneralLedgerSubAccount business class so that it edits the value of a subaccount to make sure that it only contains numeric characters.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Background for Exercise 7.4

Company XYZ wants to be able to delete certain records but cannot because the records are marked as used by a transaction. The Used By A Transaction field, a Boolean field, is typically not visible by users. To get around this, the company wants you to create a new user action to reset the Used By A Transaction field on a record from True to False so that the record can be deleted. This type of user action is a Function action.

Note: You need to be careful when using a Function action, because you can do anything within the action and could accidentally cause data to be corrupted. For any user-defined action, it is advisable to consult with Infor consultants to discuss what the action should do and review your code before you implement.

Exercise 7.4: Create a user action

- In this exercise, you will create a new action, one that was not delivered with the application.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Background for Exercise 7.5

The UniqueJournalID field in the journal header for the GeneralLedgerJournalControl business class ensures that the same journal is not used more than once. If a value is entered in this field, it must be unique.

Your organization wants to use the same value for UniqueJournalID every month. So instead of it being a unique ID throughout the system for all periods, they want it to be unique for just for one period.

To do this, you need to configure a derived field to concatenate the UniqueJournalID value with the AccountingEntity, and the year, month and day from the PostingDate. Then, you will need to configure the CreateUnreleased action to replace the value of UniqueJournalID with the new concatenated value in the Entrance Rules, if the UniqueJournalID is entered.

Exercise 7.5: Modify how the UniqueJournalID field functions

- In this exercise, you will string together values in a derived field, extract parts of a date, and code entrance rules in a configured action. You will add a configuration to the Create action of the GeneralLedgerJournalControl.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Background for Exercise 7.6

A client wants to allow a comment to be added when creating a new journal in the GeneralLedgerJournalControl business class.

First, you will create two transient user fields on the GeneralLedgerJournalControl business class – Title (alpha 50) and Comment (text).

Next you will configure the CreateUnreleased action on the GeneralLedgerJournalControl which will create a GeneralLedgerJournalComment record in the exit rules if the new transient fields were entered. If the Comment field is entered, the Title field must also be entered and vice versa.

Finally, you will configure the GeneralLedgerJournalControl form to include those user fields, but are only visible when the journal does not exist (based on the GeneralLedgerJournalControl field being blank).

Exercise 7.6: Use the ‘Visible When’ LPL syntax to add a comment when creating a new journal

- In this exercise, you will configure the CreateUnreleased action on the GeneralLedgerJournalControl business class. You will create two required transient fields to store data that will in turn create a comment record in the GeneralLedgerJournalComment business class. This is an exit rule because it does not fire until after the rest of the code runs.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Background for Exercise 7.7

A client wants to update the journal ledger application so that supporting documentation must be attached to a journal entry. If there is no supporting documentation, the journal cannot be released.

First, you will configure the action - GeneralLedgerJournalControl.ReleaseUnreleased – to have a constraint that checks for the presence of attachments in the related GeneralLedgerJournalDocument business class. Later you will test the configuration.

Exercise 7.7: Attach supporting documentation

- In this exercise, you will use a built-in relation to check for the existence of data in the entrance rules of a configured action.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Background for Exercise 7.8

A client wants to ensure that dimension values do not contain any embedded spaces.

First you will define a transient user field that will be used as a subscript Next, you will configure the Create action for FinanceDimension1, adding logic in the entrance rules to loop through the characters of the dimension value editing to make sure there are no embedded spaces in the FinanceDimension1 field.

Exercise 7.8: Ensure there are no embedded spaces in a dimension value

- In this exercise, you will check the size of a field, code a while loop, and inspect characters using a subscript.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Demo 7.9: Free-form text in fields

Your instructor will demonstrate that users can enter free-form text into the BankEntity fields on the Vendor form.

The purpose of this demo is to show that you may want to prevent users from entering free-form text in certain fields.

Background for Exercise 7.10

The BankEntity and IntermediaryBankEntity fields on the Vendor business class implements a pattern that allows a user to select from a list of values, or type anything manually in the free form field. A client wants to require that values exist in the system and free form entries are not allowed.

Exercise 7.10: Prevent free-form entry of invalid values into fields

- In this exercise, you will add a configuration to the entrance rules of the Vendor Update action. You will also work with key field existence checks.
- Observe as your instructor first demonstrates this exercise.
- Refer to your course materials for instructions to complete this activity on your own.

Check your understanding

Check your understanding

Name at least three of the six delivered actions.

- Create
- Update
- Delete
- Instance
- Set
- Purge

Check your understanding

How do action requests differ from actions?

Action requests require that you select a process flow service to control the process of routing the action request to approvers. Action requests also may require selection of more parameters and parameter rules to accommodate possible contingencies.

Check your understanding

Where can you add entrance and exit rules?

You can define entrance and exit rules for delivered LPL actions. They can access additional services outside of Landmark applications.

You cannot add entrance or exit rules for business classes that are stored in the GEN Landmark environment.

Course summary

Learning objectives

- Now that you have completed this course, you should be able to:
  - Describe Landmark Pattern Language (LPL).
  - Describe Landmark ontology.
  - Configure a business class.
  - Create a user business class.
  - Create user fields.
  - Configure common user interface objects.
  - Configure an action.

Course review

Check your understanding

From Infor Rich Client, how do you access Configuration Console to configure an application?

Start > Configure > Application

Check your understanding

What are three ways to access LPL code?

LPL Editor (pencil), Configure link, and LPL Viewer

Check your understanding

Which two of the following statements about the Landmark ontology model are true?

a) It defines the order in which things must exist to have meaning.

b) It creates artificial relationships between objects.

c) Once created, the structure is permanent.

d) It is based on two primary objects: business classes and key fields.

Check your understanding

If the LPL syntax for a field contains a return statement, what kind of field is it defining?

a) Condition
b) Compute
c) Derived
d) User

Check your understanding

What is a delivered business class?

Business classes contain the basic definitions for what data is in an application, how it is organized, and how it is processed.

A delivered business class is one that Landmark provides to cover common business needs. It is ready to use out of the box or can be configured.

Check your understanding

Which two of the following are defined in the LPL for a menu item?

a) Name of the menu item
b) Color and font of the menu item
c) Object accessible from the menu item
d) Security rules for the menu item

Additional resources

Documentation

- Access Infor Concierge for support materials and related information at the following web https://concierge.infor.com

Training courses

- Access Infor Campus for a complete listing of course offerings for this product line at the following web address: http://campus.infor.com

Whiteboard

Infor Education
Master Infor products.
Maximize your potential.

Thank you