Lawson: Customizing Lawson Applications with Design Studio and JavaScript Training Workbook
Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade, or service names referenced may be registered trademarks or trademarks of their respective owners.
Table of contents

About this workbook .................................................................................................................. 5
Course overview .......................................................................................................................... 6
Course description and agenda ................................................................................................. 7

Lesson 1: Infor Lawson Design Studio overview ..................................................................... 9
  Infor Lawson Design Studio components ............................................................................. 10
  Infor Lawson Design Studio interface ................................................................................ 11
  Infor Lawson Design Studio menu system ........................................................................... 12
  Infor Lawson for Infor Ming.le (Portal) ............................................................................... 13
  Infor Lawson Design Studio directory structure .................................................................. 15
  Terminology .......................................................................................................................... 16
  Check your understanding .................................................................................................... 18

Lesson 2: Wizard Designer ..................................................................................................... 20
  Wizard Designer overview .................................................................................................. 21
  Check your understanding .................................................................................................... 27

Lesson 3: User Interface (UI) Designer ................................................................................... 28
  UI Designer overview ......................................................................................................... 29
  UI Designer design window ................................................................................................. 30
  UI Designer objects ............................................................................................................ 32
  Shortcuts ............................................................................................................................... 41
  Check your understanding .................................................................................................... 47

Lesson 4: Custom Page Designer ........................................................................................... 49
  Custom Page Designer overview ....................................................................................... 50
  Custom Page Designer objects .......................................................................................... 51
  Querying Infor Lawson data .............................................................................................. 60
  Mapping data query objects .............................................................................................. 64
  Check your understanding .................................................................................................... 72

Lesson 5: Enhancing Infor Lawson Design Studio with JavaScript ........................................ 74
  JavaScript overview ........................................................................................................... 75
  Custom scripting in Infor Lawson Design Studio ................................................................. 76
  Working with multiple forms .............................................................................................. 91
  Enhancing a custom page with JavaScript ......................................................................... 98
  Check your understanding .................................................................................................... 104

Course summary .................................................................................................................... 106
Course review ......................................................................................................................... 107
About this workbook

Welcome to this Infor Education course! We hope you will find this learning experience enjoyable and instructive. This Training Workbook is designed to support the following forms of learning:

- Classroom instructor-led training
- Virtual instructor-led training
- Self-directed learning

This Training Workbook is not intended for use as a product user guide.

Self-directed learning (SDL)

If this course is eligible for self-directed learning, demos and exercises throughout this Training Workbook will be hyperlinked to Demonstration/Let Me Try simulations that allow you to view and practice the execution of the demo or exercise in a simulated training environment.

Activity data

You will be asked to complete some practice exercises during this course. Step-by-step instructions are provided in this guide to assist you with completing the exercises. Where necessary, data columns are included for your reference.

Your instructor will provide more information on systems used in class, including server addresses, login IDs, and passwords.

Symbols used in this workbook

- Hands-on exercise ("Exercise")
- For your reference
- Question

- Instructor demonstration ("Demo")
- Your notes
- Answer

- Can be used for either "Scenario" or "Discussion"
- Note
- Task simulation
Course overview

Reference materials
Infor® Lawson Design Studio reference materials are available from the following locations:
- Infor Lawson Design Studio Help menu
- Infor Xtreme®

Course duration
24 hours

Learning objectives
Upon completion of this course, you will be able to:
- Define the process for creating help text and “stepping” users through forms using Wizard Designer.
- Identify the process for customizing Infor Lawson forms using User Interface (UI) Designer.
- Define ways to allow automation of form actions such as add, delete, and inquire.
- Describe the process for using the Custom Page Designer to create custom pages.
- Explain how to use Infor Lawson API Builder to build the string used to retrieve and send Infor Lawson data through the Internet Object Services (IOS) engine.
- Define how to extend the functionality of a form using JavaScript™.
- Describe how to use Infor Lawson Design Studio’s JavaScript development environment to add customized functionality to a form or custom page

Audience
- Customer User
- Pre-Sales Consultant
- Business Consultant
- Technical Consultant
- Support
- System Administrator

System requirements
- EDU: LSF 10 DS Training Environment
Course description and agenda

This course covers skills and knowledge around Infor Lawson Design Studio V10, a browser based drag and drop graphical development environment which can be used to customize or enhance Infor Lawson applications.

**Prerequisite knowledge**

To optimize your learning experience, Infor recommends that you have the following knowledge prior to attending this course:

- General knowledge of Infor Lawson applications and user interface.
- General understanding of JavaScript is desired.

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Lesson title</th>
<th>Learning objectives</th>
<th>Day</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Course overview</td>
<td>• Review course expectations.</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>Infor Lawson Design Studio overview</td>
<td>• List the components of the Infor Lawson Design Studio application.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Identify sub menus and their uses.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Identify the types of views and their uses.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Wizard Designer</td>
<td>• List the steps to build a wizard using Wizard Designer.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Identify the elements of the Wizard Designer interface.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>UI Designer</td>
<td>• Explain how to use the User Interface (UI) Designer to customize Infor Lawson application forms.</td>
<td>1 and 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Identify the types of customizations possible with the UI Designer.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Recognize the differences between the various types of form objects.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Explain how to extend self-service with Infor Lawson Design Studio.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Custom Page Designer</td>
<td>• Identify types of objects and their uses within Custom Page Designer.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Describe the process for building a data query object and custom menu.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Define the steps to map both data and non-data objects.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Enhancing Infor Lawson Design Studio with JavaScript</td>
<td>• Define how to extend functionality of a form using JavaScript.</td>
<td>2 and 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Describe the steps to designate a required field.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Define ways to limit the options available in a user’s selection list.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Describe how to combine multiple functions using JavaScript.</td>
<td></td>
</tr>
<tr>
<td>Lesson</td>
<td>Lesson title</td>
<td>Learning objectives</td>
<td>Day</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• List the steps to perform a calculation on a form that has been modified with a new field.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Explain the process for using Custom Page Designer to add a button on a custom page for approving invoices.</td>
<td></td>
</tr>
<tr>
<td>Course summary</td>
<td></td>
<td>• Debrief course.</td>
<td>3</td>
</tr>
</tbody>
</table>
Lesson 1: Infor Lawson Design Studio overview

Estimated time
2 hours

Learning objectives
After completing this lesson, you will be able to:

- List the components of the Infor Lawson Design Studio application.
- Identify sub menus and their uses.
- Identify the types of views and their uses

Topics
- Infor Lawson Design Studio components
- Infor Lawson Design Studio interface
- Infor Lawson Design Studio menu system
- Infor Lawson for Infor Ming.le (Portal)
- Infor Lawson Design Studio directory structure
- Terminology
- Check your understanding
Infor Lawson Design Studio components

Infor Lawson Design Studio™ is a browser based drag and drop graphical development environment which can be used to customize or enhance Lawson applications in a variety of ways. The three components and their features are shown in this image.

UI Designer
• Customizes Infor Lawson forms
• Creates new forms

Wizard Designer
• Allows automation of form actions
• Creates help text specific to your site

Custom Page Designer
• Creates custom pages to access Infor Lawson and other sites

Additional Features: Lawson Application Program Interface (API) Builder and JavaScript™ scripting environment

**Additional features:**

- **Infor Lawson Application Program Interface (API) Builder:** The Infor Lawson API Builder retrieves and sends Infor Lawson data through the Internet Object Services engine. API strings created using the API builder can be pasted into other applications.

- **Scripting environment:** Infor Lawson Design Studio includes a JavaScript™ development environment for adding functionality specific to your needs to a form or custom page. Within the environment, Object Viewer is provided that makes it easy to understand and access Infor Lawson Design Studio objects when you create a script.

**IOS**

Internet Object Services (IOS) serves as the conduit between Infor Lawson’s web products and the Infor Lawson database. The three main IOS services are:

- **Transaction Service (previously Application Gateway Services or AGS):** This is an API string that calls read or write data to or from Infor Lawson business logic. Typical user requests that utilize a transaction call are add, change, inquiry, delete, next, and previous.

- **Data Service (previously Data Mining Engine or DME):** This is an API string that calls the database directly. A Data call dynamically selects records from the Infor Lawson database. Typical user requests that utilize a data call are query strings (data queries) against database tables.

- **Drill Service (previously Internet Drill Around or IDA):** This is an API call that reads data using predefined Drill Around screen rules and object rules when users request a Drill Around or Select list.
Infor Lawson Design Studio interface

Logging on to Infor Lawson Design Studio

Access Infor Lawson Design Studio using your browser to navigate to the location where the application is installed. For example: http://MyServer/lawson/studio

During this training, you will use the following server: http://lsf10.gdeinfor2.com/lawson/studio

Your instructor will give you a username and password for logging on to the training environment.

Demo: Logging on to Infor Lawson Design Studio

Your instructor will demonstrate how to log on to Infor Lawson Design Studio.

Demo steps

1. Double-click the Design Studio shortcut on the training desktop. The Infor Lawson Design Studio login screen opens.
2. Type lawson in the Username field. Note: To prepare any field in the system for text entry, first click in the text field.
3. Type Infor08 in the Password field.
4. Click Login. The Design Studio Files menu opens.
Infor Lawson Design Studio menu system

Infor Lawson Design Studio is a multiple-window development environment. This means you can open as many documents ( wizards, forms, or custom pages) as your system’s memory can accommodate.

**Main menu system**

The main menu that appears along the top of all Infor Lawson Design Studio windows contains the following submenus:

<table>
<thead>
<tr>
<th>Sub-menu</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>The File sub-menu contains typical file management options, including New, Open, Save, Save As, and Close. The File menu includes a “Recent” option that lists the four most recently accessed files.</td>
</tr>
<tr>
<td>Window</td>
<td>The Window sub-menu expands or collapses all active windows. The active window is the one that is currently “on top” and is highlighted.</td>
</tr>
<tr>
<td>Edit</td>
<td>The Edit sub-menu includes Undo and Redo as well as an option to Select or Deselect items. Shift Upwards and Shift Downwards are available within the Wizard Designer. Apply Changes option is available when working within the scripting window.</td>
</tr>
</tbody>
</table>
| View     | The View sub-menu provides alternative ways to access Infor Lawson Design Studio views. Some of these options are available as tabs along the menu bar that runs along the bottom of the UI Designer and Custom Page Designer windows. View options include:  
  **Toolbox** – This view contains objects that you add or modify when you create forms and custom pages.  
  **Properties pane** – This view displays the properties you can change for a selected object.  
  **Design** – This view provides access to toolbox menu and properties pane in order to make changes to a form or a custom page. Design is the default view when working in UI Designer or Custom Page Designer.  
  **Object** – This view is used for complex objects such as tab areas and detail areas that have parent-child relationships.  
  **Source** – This view shows the XML source behind selected custom forms or custom pages. Changes must be applied to take effect in Source view.  
  **Script** – This view is used for adding JavaScript to a form or custom page. |
| Tools    | The Tools sub-menu provides access to some software tools including:  
  **API Builder** – This tool is used for creating API calls to retrieve data from Infor Lawson engines.  
  **Rebuild Forms Index** – This tool is used to rebuild the index for forms in your data area. You must rebuild the index for forms in your data area when you add new forms to a data area.  
  **Set Forms Data Area** – This tool is used to change a data area for a form.  
  **Set Wizards Data Area** – This tool is used to change a data area for a wizard. |
Infor Lawson for Infor Ming.le (Portal)

Infor Lawson for Infor Ming.le™ (previously called Portal) is one access point to Infor Lawson applications and data from within the Infor system. This is the recommended user interface to use during this training course. In Infor Lawson Design Studio 10.x, previewing is done via Infor Lawson for Infor Ming.le.

To log on to Infor Lawson for Infor Ming.le, navigate to the user interface, and type your username and password into the appropriate fields. Please follow along as your instructor demonstrates how to log on.

Infor Lawson for Infor Ming.le home page
The home page is the first page that appears when you log on to Infor Lawson for Infor Ming.le.

Main toolbar
The toolbar comprises the items that appear along the top of Infor Lawson for Infor Ming.le directly underneath the Infor logo. There are items on the toolbar that are always available regardless of the action you are currently performing. The toolbar always contains the Home, Help, Preferences, and the search field.

Search field
The search field is a textbox that is always available on the toolbar. You can use it to search for Infor Lawson applications and data.

To search, you may type as much information as you have about the item and then click the spyglass icon or press Enter. You may also filter your search to include just bookmarks or applications.

Application form
You can access an application form by typing the form name or form code into the search field or by clicking a bookmark.

Demo: Log on to Infor Lawson for Infor Ming.le, navigate the toolbar, search for an item, and bring up a Lawson form
Your instructor will demonstrate how to navigate and work within the Infor Lawson for Infor Ming.le application.

Demo steps
1. Double-click the Portal shortcut on the training desktop. The Infor Lawson for Infor Ming.le login screen opens.
2. Type lawson in the Username field.
3. Type Infor08 in the Password field.
4. Click Login. The Infor Lawson for Infor Ming.le home page opens.
5. Click the various tabs in the toolbar.
6. Click the Search box (upper right hand corner).
7. Type AP10.1 to access a Lawson form.
Infor Lawson Design Studio directory structure

<table>
<thead>
<tr>
<th>Item</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Infor Lawson Design Studio content directory</td>
<td>LAWDIR/persistdata/lawson/portal/content</td>
</tr>
<tr>
<td>UI Designer form customizations</td>
<td>lawdir/persistdata/lawson/portal/content/forms</td>
</tr>
<tr>
<td>Custom Page Designer pages</td>
<td>lawdir/persistdata/lawson/portal/content/pages</td>
</tr>
<tr>
<td>(subdirectories exist for each object type)</td>
<td></td>
</tr>
<tr>
<td>Wizard Designer wizards</td>
<td>lawdir/persistdata/lawson/portal/content/wizards</td>
</tr>
<tr>
<td>Images/logos</td>
<td>lawdir/persistdata/lawson/portal/content/images</td>
</tr>
<tr>
<td>Reports</td>
<td>lawdir/persistdata/lawson/portal/content/reports</td>
</tr>
<tr>
<td><strong>JavaScript</strong></td>
<td></td>
</tr>
<tr>
<td>JavaScript files</td>
<td>lawdir/persistdata/lawson/scripts</td>
</tr>
</tbody>
</table>
## Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Object Services (IOS)</td>
<td>Internet Object Services serves as the conduit between Infor Lawson’s web products and the Infor Lawson database.</td>
</tr>
<tr>
<td>Portal (generic term, not Infor specific)</td>
<td>A portal is a web site or service that offers a variety of resources and services, such as e-mail, forums, search engines, and online shopping malls.</td>
</tr>
<tr>
<td>Thin client</td>
<td>A thin client requires only a browser to run an application from a server.</td>
</tr>
<tr>
<td>Browser</td>
<td>A browser is also known as a web browser, which can serve as a user interface to an application</td>
</tr>
<tr>
<td>XML</td>
<td>XML is a language used to define the structure of documents.</td>
</tr>
<tr>
<td>Wizard</td>
<td>Wizard is a utility attached to an Infor Lawson form that provides step-by-step instructions on how to complete a specific task.</td>
</tr>
<tr>
<td>Application Program Interface (API) Builder</td>
<td>The API Builder is a tool that lets you point and click through a series of form fields to generate a query statement for transaction, data, and drill calls. This tool is used for creating API calls to retrieve data from Infor Lawson engines.</td>
</tr>
<tr>
<td>Transaction Service (previously AGS)</td>
<td>The AGS is an API string that calls read or write data from or to an Infor Lawson application. Typical user requests that use this string are add, change, delete, inquire, next, and previous.</td>
</tr>
<tr>
<td>Lawson Portal (now called Infor Lawson for Infor Ming.le)</td>
<td>This portal provides links to Infor Lawson forms or web pages and displays information from an Infor Lawson application.</td>
</tr>
<tr>
<td>Web server</td>
<td>The software program that resides on a server that delivers or serves up web pages.</td>
</tr>
<tr>
<td>Application Program Interface (APIs)</td>
<td>APIs are calling conventions that let programs communicate between three tiers of Infor Lawson.</td>
</tr>
<tr>
<td>Portal objects (now called Custom Page Designer objects)</td>
<td>Portal (Custom Page Designer) objects are used in building a portal (custom page). Infor Lawson Design Studio provides an extensive list of these, such as web pages, menus, images, reports, and text.</td>
</tr>
<tr>
<td>Data Service (previously DME)</td>
<td>A data service is an API program that is used to select records of data from an Infor Lawson database.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Drill Service (previously IDA)</td>
<td>A drill service is an API program that reads the database for Drill Around and Select requests.</td>
</tr>
<tr>
<td>Custom page</td>
<td>A custom page is created within the Custom Page Designer, consisting of content from custom objects.</td>
</tr>
</tbody>
</table>
Check your understanding

_________ is an Infor Lawson Design Studio component that lets users customize Infor Lawson application forms.

a) Custom Page Designer  
b) Wizard Designer  
c) UI Designer

_________ creates a string to retrieve and send Infor Lawson data through the IOS engines.

a) JavaScript  
b) API Builder  
c) Portal

_________ is one access point to Infor Lawson applications and data from within the Infor system.

a) Infor Lawson Design Studio  
b) Properties pane  
c) Infor Lawson for Infor Ming.le (Portal)

The ___________ provides access to the toolbox menu and properties pane in order to make changes to a form or a custom page.

a) Objects view  
b) Design view  
c) Script view
The _______________ is a component of Infor Lawson Design Studio that lets users create new or modify current content within the Infor Lawson for Infor Ming.le.

a) UI Designer  
b) Wizard Designer  
c) Custom Page Designer

Identify the three main IOS services.

a) Transaction Service  
b) Data Service  
c) API  
d) Drill Service
Lesson 2: Wizard Designer

Estimated time
2 hours

Learning objectives
After completing this lesson, you will be able to:

- List the steps to build a wizard using Wizard Designer.
- Identify the elements of the Wizard Designer interface.

Topics
- Wizard Designer overview
- Check your understanding
Wizard Designer overview

The Wizard Designer is a tool that automates form actions such as inquire, add, and open a sub form. It can combine an automated form action with instructions and create a set of instructions for users. Wizard Designer is similar to online help but customized to your site.

In addition to letting you add site-specific information to a form, a wizard can step a user through a form. You can program the wizard to ensure that the user performs required steps or, when possible, perform the steps for the user.

Wizard Designer interface

The Wizard Designer interface includes the following elements:

- Wizard ToolBox
  - Includes: Step, Branch, and Wizard
- Properties pane
  - Used for changing properties related to the currently active tool

Building a wizard

Wizards are made up of steps. You create steps and associate them with fields. A step can include a condition statement or a branch. When a branch exists for a step, the wizard will carry out a step or steps based on data that is typed in the field.

Text that you want to add to a step are added from the bottom panel of the main Wizard Designer window. As a default, the Infor Lawson field help for the current field appears.

You can add all the fields at once to the wizard, then add text to each, or you can add a field at a time, completing the help text before selecting another field. **Hint:** Press and hold the CTRL key to select multiple objects at one time.

Once created, you cannot change the file name of a wizard.

Wizards can be helpful to remind users how to perform tasks that they may only do once a year, such as setup required for annual benefits enrollment.
Demo: Build a wizard for the Salary Rate Table (BN02.2) form

Your instructor will demonstrate how to build a wizard for the Salary Rate Table (BN02.2) form.

Exercise 2.1: Build a wizard for the Salary Rate Table (BN02.2) form

In this exercise, you will build a wizard for the Salary Rate Table (BN02.2) form that will help the Benefit Clerk complete a task to update a flex dollar calculation.

Exercise 2.1 steps

Part 1: Build a wizard

1. Double-click the Design Studio shortcut on the desktop. The Infor Lawson Design Studio login screen opens.
2. Type lawson in the Username field.
3. Type infor08 in the Password field.
4. Click Login. The Design Studio Files menu opens.
5. Select Wizard Designer.
6. Click OK. The New Wizard dialog window opens.

Notes:
- You can also double-click Wizard Designer to launch the Wizard Designer form.
- The New Wizard dialog window is used to select data area, system code, and form for the new wizard.
7. Select APPS10.
9. Select BN (Benefits).
10. Click Next. The Form Selection dialog window opens.
11. Select BN02.2 (Salary Rate Table).
12. Click Next. The Confirm Selections dialog window opens.
13. Make sure the selections made in steps 7, 9, and 11 display on the Confirm Selections window.
14. Click Finish. The main Wizard Designer window now appears on the screen.
Part 2: Rename wizard0 to “Add”

1. Select the wizard0 folder. The Name field in the right-hand Properties pane auto fills with this selection.
2. Type Add in the Name field on the Properties pane. Notice how the folder is now renamed Add in the main Wizard Designer window.

Part 3: Add steps to the Add wizard

1. Select Start Wizard.
2. Double-click Step on the Wizard ToolBox. The Field dialog window opens.

Notes:
- Each time you double-click Wizard, Branch, or Step, a drop-down list appears in the dialog window from which you can make your selections to add to the wizard.
- Items display on the main Wizard Designer design window after you select the option and click OK.
3. Select RTH-COMPANY from the list of Fields.
4. Click OK. The RTH-COMPANY field is added to the main Wizard Designer window.
5. Select RTH-COMPANY in the main Wizard Designer window.
7. Select RTH-START-DATE.
8. Click OK. The RTH-START-DATE field is added to the main Wizard Designer window.

You can add all the fields at once to the wizard, then add text to each, or you can add a field at a time, completing the help text before selecting another field.

9. Select RTH-COMPANY.
10. Click Clear in the Step Text section. The existing help text is deleted from the window. Note: The Step Text section appears at the bottom of the main Wizard Designer window.
11. Type Enter company value 4321 in the Step Text free-flow text entry field. Help text is added to the RTH-COMPANY field.
12. Select RTH-START-DATE.
13. Highlight Type the start date of the rate table in the Step Text section.
15. Type Enter either January 1 or October 1 of the next plan year in the Step Text free-flow text entry field.

Part 4: Add a branch to the RTH-START-DATE field

1. Select RTH-START-DATE in the main Wizard Designer window.
3. Type 01/01/2015 in the Condition field.
4. Click OK. The date 01/01/2015 is added in the main Wizard Designer window.
5. Select 01/01/2015.
7. Select RTD-BEG-AGE-YR-SAL.
8. Click OK. The RTD-BEG-AGE-YR-SAL field is added to the main Wizard Designer window.
9. Select RTH-START-DATE.
10. Double-click Branch.
11. Type 10/01/2015 in the Condition field.
12. Click OK. The 10/01/2015 date is added to the main Wizard Designer window.
13. Select 10/01/2015.
15. Select RTD-EMP-RATE.
16. Click OK. The RTD-EMP-RATE field is added to the main Wizard Designer window.
17. Double-click Step.
18. Select RTD-COMP-RATE.
19. Click OK. The RTD-COMP-RATE field is added to the main Wizard Designer window.
20. Select End Wizard.

Part 5: Save the wizard
1. Select File > Save As. The Save dialog window opens.
2. Type bn02.2.xml in the File name field.
3. Click OK. The bn02.2.xml file is saved.

Part 6: Preview and save the wizard
1. Select the Add folder on the main Wizard Designer window.
2. Select View > Preview. A preview of the BN02.2 (Salary Rate Table) form opens. Note: Previewing the wizard is done using the Infor Lawson for Infor Ming.le (Portal).
3. Click Continue. The wizard begins.
4. Click Continue to move through the fields.
5. Click X to close the preview window and return to Wizard Designer.

Adding another wizard to a form
A form can be used for more than one purpose, therefore, you can create multiple wizards for a single form. When doing so, you must ensure to create the new form in such a way that you do not overwrite the existing wizard.

Verify if a wizard already exists for a form
Check the lawdir/persistdata/lawson/portal/content/wizards directory to see if an XML definition file already exists for a wizard.

When adding a new wizard to a form with an existing wizard, be sure to position your cursor after the last step in the current wizard and then click the Wizard tool.
When a wizard already exists for a form, do not use the Wizard Designer’s File > New option in Infor Lawson Design Studio; this will overwrite the existing wizard. Instead, use the Wizard tool to create the new wizard.

Demo: Add another wizard to a form
Your instructor will demonstrate how to add another wizard to a form.

Exercise 2.2: Add another wizard to a form
In this exercise, you will add an inquire action wizard to the BN02.2 form.

Exercise 2.2 steps

Part 1: Add Inquire wizard
1. Open the bn02.2.xml wizard you created in exercise 2.1, part 5. Note: If prompted to select between Remote or Local to save your wizard, select Remote.
2. Click OK.
3. Set your cursor after the last step in the current wizard, for example, End Wizard.
4. Double-click Wizard in the Wizard ToolBox. A new wizard0 folder displays on the same form and steps can be added.

Part 2: Rename the wizard0 to Inquire
1. Select the wizard0 folder.
2. Type Inquire in the Name field on the Properties pane. The folder, now renamed Inquire, displays in the main Wizard Designer window.

Part 3: Add steps to the Inquire wizard
1. Select Start Wizard below the Inquire folder on the main Wizard Designer window.
2. Double-click Step on the Wizard ToolBox.
3. Select RTH-COMPANY.
4. Click OK.
5. Double-click Step.
6. Select RTH-TABLE-CODE.
7. Click OK.
8. Double-click Step.
9. Select RTH-START-DATE.
10. Click OK.
11. Select End Wizard.
12. Click Action (…).
13. Select Inquire (I).
14. Click OK.

Part 4: Save the wizard
1. Select File > Save. **Note:** If prompted to select between Remote or Local to save your wizard, select Remote.
2. Select bn02.2.xml from the file names, if prompted.
3. Click OK. The XML file is saved.
4. Click OK if prompted to overwrite the existing file.
5. Click X to close Infor Lawson Design Studio.

Part 5: Preview the wizard. **Note:** This method will allow you to view and choose between both Wizard Designer options for bn02.2.
1. Double-click the Portal shortcut on the training desktop. Infor Lawson for Infor Ming.le opens.
2. Type bn02.2 in the Search field.
3. Press Enter.
4. Click Help (?) on the Infor Lawson for Infor Ming.le (Portal) home page.
5. Select Form Wizard. A pop-up box opens displaying all available wizards for the form.
7. Click Continue to move through the fields.
8. Click X to close Infor Lawson for Infor Ming.le.

When you create an additional wizard for a form where at least one wizard already exists, you must take care to create the new form in such a way that it does not overwrite the existing wizards.
Check your understanding

A _______ is a utility attached to an Infor Lawson form that provides step-by-step instructions on how to complete a specific task.
   a) Wizard Designer
   b) Menu
   c) API

Which two elements are part of the Wizard Designer interface?
   a) Properties pane
   b) Toolbox
   c) Portal page

The Wizard Designer utilizes __________ and associates them with fields to build wizards.
   a) Scripts
   b) Steps
   c) Menus

True or False: A wizard must be saved as the Infor Lawson form name and cannot be changed.
   a) True
   b) False

The Wizard Designer can automate form actions such as: Select all that apply.
   a) Add
   b) Inquire
   c) Open

Add a _______ for a step, if you want the wizard to carry out a step or steps based on data that is typed in the field.
   a) Field
   b) Branch
   c) Label
Lesson 3: User Interface (UI) Designer

Estimated time
6 hours

Learning objectives
After completing this lesson, you will be able to:

- Explain how to use the UI Designer to customize Infor Lawson application forms.
- Identify the types of customizations possible with UI Designer.
- Recognize the differences between the various types of form objects.
- Explain how to extend self-service with Infor Lawson Design Studio.

Topics
- UI Designer overview
- UI Designer design window
- UI Designer objects
- Shortcuts
- Check your understanding
UI Designer overview

The UI Designer component of Infor Lawson Design Studio is used to modify or customize the original Infor Lawson-delivered version of a form.

You can use UI Designer to:

- Change the tabbing order on a form
- Remove tabs from a form with multiple tabs
- Remove fields
- Add images or logos to a form
- Expose hidden fields
- Change selection criteria for a value list
- Create a new form customization using a blank palette

Your new form will exist in addition to the Infor Lawson-delivered version of the form. You will not be allowed to overwrite the Infor Lawson version of the form.
UI Designer design window

When you create a new form customization, you are prompted to select data area, system code, form, QuickPaint, or Blank Palette.

If you select QuickPaint, the form appears in the design window with all the fields and other objects from the original Infor Lawson form in place. You would use QuickPaint when you are making only minor changes to the form. To add objects to the form, double-click the object from the UI Toolbox. Once added to the form, select the object to view its associated properties in the Properties pane.

If you select Blank Palette, the design window is blank. No fields or other objects from the original Infor Lawson form appear in the window. All functionality of the form is available, but you must add each object you want to appear on the form. When you are significantly modifying an Infor Lawson form (for example, removing or rearranging many of the fields), starting from a blank form might be easier than QuickPaint.

Before beginning your UI Designer work, it’s important to determine whether or not you will use a QuickPaint screen or Blank Palette. To do so, ask yourself, “Will I be taking off more than I’m leaving on the form?” If the answer is yes, you will want to begin with Blank Palette.

The UI Designer used alone (without making use of JavaScript) can alter only functionality that a form already has. It cannot add functionality.

Demo: Select a QuickPaint form
Your instructor will demonstrate how to select a QuickPaint form.

Exercise 3.1: Select a QuickPaint form
In this exercise, you will select a QuickPaint form.

Exercise 3.1 steps
1. Double-click the Design Studio shortcut on the training desktop. The Infor Lawson Design Studio login screen opens.
2. Type lawson in the Username field.
3. Type Infor08 in the Password field.
4. Click Login. The Design Studio Files window opens
5. Select UI Designer.
6. Click OK. The New User Interface Wizard dialog window opens.
7. Select APPS10 in the Data Area Selection section.
8. Click Next.
10. Click Next.
12. Click Next.
13. Select QuickPaint. Make sure the selections for data area, system code, and form identification (Id) appear in the Confirm Selections window.
14. Click Finish. The User Interface Designer window opens with a QuickPaint form.
15. Click the Source tab. The Source tab is located in the bottom of the User Interface Designer window. Note: You do not have to change the code displayed, however, you can view it to see how the form is built.
   
   Note: Keep the FR10.1 (Customer Contract) form open to use in the next exercise.

UI Designer and Custom Page Designer have a view menu bar that runs along the bottom of the design window. When you click Design, Object, Source, or Script, your working environment changes accordingly.
UI Designer objects

Infor Lawson Design Studio User Guide

Toolbox and Properties pane

The UI Designer, like the Wizard Designer, displays a Toolbox pane on the left-side of the User Interface Designer window and a Properties pane on the right-side.

The UI Toolbox includes objects such as radio buttons, labels, and text input boxes that allow users to select or add data to the form as well as being able to use graphical elements such as lines and font color to add visual interest to the design. Each object has a set of associated properties. Properties are aspects or values of the object that you can change. Once objects are added to the form, you can select it to customize its properties or rearrange how it is displayed on the form using the Properties pane. The properties that are available for customization vary depending on the selected object.

Some properties require the user to select from a drop-down list, while other properties require the user to type a value into a field. For example, the form object Textbox has a Tooltip property that allows the user to type onto a textbox.

Types of UI Designer form objects

- User input boxes include buttons, radio buttons, checkboxes, textboxes, and listboxes.
- Design objects include labels, rectangles, lines, and images.
- Special form objects include forms, tab areas, detail areas, and browsers.

Example of toolbox objects uses:

- Defaulting field values
- Field validation
- Fields from another form
- Changing the layout of a field
- Microsoft® Explorer type of view
- Multiple versions of one form
- Automatically fill fields
- Automating functionality
- Limiting data selection
- Different view of a form
Demo: Customize design objects for Customer Contract (FR10.1) form using QuickPaint

Your instructor will demonstrate how to customize design objects for the Customer Contract (FR10.1) form using QuickPaint.

Exercise 3.2: Customize design objects for Customer Contract (FR10.1) form using QuickPaint

In this exercise, you will customize design objects for the Customer Contract (FR10.1) form using QuickPaint.

Exercise 3.2 steps

Part 1: Remove fields and buttons not needed on the FR10.1 form

1. Click the Design tab at the bottom of the User Interface Designer window. Note: You will be working with the open FR10.1 form in the User Interface Designer window from exercise 3.1.

2. Select the following textboxes and their corresponding labels on the form. Note: To select a textbox and corresponding label highlight the label name and the textbox. Hint: Press and hold the Ctrl key to select multiple objects at one time.
   - Variance Limit
   - Taxable
   - Budget
   - Group Code
   - Estimates
   - Require Aggregate Sales
   - Aggregate Sales by Date

3. Select Edit > Delete. The textboxes and corresponding labels are removed from the form.

4. Delete the following buttons. Note: To delete a button select the buttons to be removed, and then press Delete.
   - Budget
   - User Field

Deleting or removing fields on a form is akin to hiding them. The fields are not permanently eliminated and can be added back to the form for future use.
Part 2: Design the visual display of the remaining fields on the form by adding:

- A Required Fields section
- An Additional Information section
- A line to separate the two sections on the page

1. Select the following required fields. **Hint:** Remember you can press and hold the Ctrl key to select multiple objects at one time.
   - Company
   - Customer
   - Ship To
   - Contract

2. Drag the fields down the equivalent of the height of one field on the form. **Note:** You will add a label titled *Required Fields* later in the exercise. Make sure you leave enough room for the label.

3. Select the remaining fields and arrange them below the *Required Fields* section. **Note:** You will add a label titled *Additional Information* later in the exercise.

4. Double-click **Line** in the **UI Toolbox**. The **Line** object is added to the form.

5. Select the **Line** object.

6. Drag the **Line** object to above the **Required Fields** section.

7. Click **Width** in the **Properties** pane.

8. Type 28 in the **Width** field. **Hint:** You can also click and drag the Line object’s handles to the left or right to adjust the width.

   The Properties pane is used to customize objects added to a new form. Each time you select an object on the form, the attributes in the Properties pane that display are relative to that object. When you click an attribute, the options available for customization are enabled and/or opens. For example, when selected, the Text property opens with the Text Entry field window that allows the user to type into a textbox to enter the appropriate data.

9. Press **Tab**.

10. Click **Text (…)** in the **Properties** pane. The **Text Entry** field opens. **Note:** You must have the Line object selected on the **User Interface Designer** window.

11. Type *Required Fields* in the textbox.

12. Click **OK**.

13. Double-click **Line** in the **UI Toolbox**.

14. Select the **Line** object.

15. Drag the **Line** object to above the **Additional Information** fields.

16. Click **Width**.

17. Type 28 in the **Width** field.

18. Press **Tab**.

19. Click **Text (…)**.

20. Type *Additional Information* in the textbox.

21. Click **OK**. The form, with the changes made, displays.
Part 3: Change Status field to a checkbox and assign a data area

1. Select the Status textbox on the User Interface Designer window.
2. Write down the data source ______________________________. The data source displays in the Properties next to the Data Src button when the Status field is selected.
3. Select Edit > Delete. This removes the Status textbox from the form.
4. Double-click Checkbox in the UI Toolbox. The Checkbox object is added to the form.
5. Select the Checkbox object.
6. Drag the Checkbox object to align with the Status label. Note: Use the arrows to adjust the width and height of the checkbox.
7. Click Data Src (…) in the Properties pane. The Data Field window opens.
8. Select the FRM-ACTIVE-FLAG data source you wrote down in step 2 from the list of available data fields.
9. Click OK. This assigns the data source to the Checkbox object.

Part 4: Preview the form

2. Select the Status checkbox to see the value and display field change from Inactive to Active.
3. Click X to close the preview form and return to the User Interface Designer window.

Part 5: Set default values

1. Select the Company textbox in the User Interface Designer window.
2. Click Default (…) in the Properties pane. The Default Value window opens.
3. Select the Literal radio button.
4. Type 1234 in the Default Value field.
5. Click OK to return to the User Interface Designer window.
7. Select the Output Only box.
8. Click Apply.
9. Click OK to return to the User Interface Designer window.
10. Select the Currency textbox.
11. Click Default (…).
12. Select the Literal radio button.
13. Type USD in Default Value field.
14. Click OK.
15. Select the Process Level textbox.
16. Click Default (…).
17. Select the Literal radio button.
18. Type 1 in the Default Value field. Note: After setting the Process Level default to 1, you will delete the Process Level field so it no longer displays on the form.
19. Click OK.
20. Press and hold the Ctrl key and select the Process Level textbox and label.
21. Select Edit > Delete.
22. Select the Termination Date textbox.
23. Click Default (…).
24. Select the Literal radio button.
25. Type 12/31/2015 in the Default Value field.
26. Click OK.
27. Select the Calendar textbox.
28. Click Default (…).
29. Select the Literal radio button.
30. Type CALENDAR 1 in the Default Value field. Note: This value includes a space.
31. Click OK to return to the User Interface Designer window.

Part 6: Create tooltips (mouse-over text) for the remaining buttons
1. Select the COPY1 button in the User Interface Designer window.
2. Click Tooltip (…) in the Properties pane. The Tooltip Entry dialog window opens.
3. Type Transfers to FR10.2. Use this to add changes from an existing customer contract or from a standard customer contract in the textbox.
4. Click OK.
5. Select the More button.
6. Click Tooltip (…).
7. Type Transfers to FR10.5. Use this to add address and licensee information for this customer in the textbox.
8. Click OK to return to the User Interface Designer window.

Part 7: Make additional graphical changes to the form (e.g., add labels, change font color, change text box)
1. Double-click Label in the UI Toolbox.
2. Select the Label object.
3. Drag the Label object below the COPY1 button.
4. Click Text (…).
5. Type This form should be completed in the textbox.
6. Click OK.
7. Select the Contract Date label.
8. Click Text (…).
9. Type Date of Contract in the textbox.
10. Click OK to return to the User Interface Designer window.
Part 8: Preview the form
2. Type 1001 in the Customer field. Notice the Company field is prepopulated with 1234, the default value you set in part 5.
3. Type 1 in the Ship To field.
4. Type 1 in the Contract field.
5. Type 1/1/2015 in the Date of Contract field.
6. Click Add. The record is added to the form.
7. Click X to close the preview form and return to the User Interface Designer window.

Part 9: Save the form
1. Select File > Save As. The Form Id Entry dialog window opens.
2. Type FR10.1_<your initials> in the textbox. Write down the Form Id _______________.
3. Click OK. A custom version of the form is saved.
4. Type fr10.1<your initials>.xml in the File name field.
5. Click OK. The XML file is saved.

Customize forms using Blank Palette
Selecting a blank form means the form will open in the design window with all form fields and objects hidden. Any objects that you want to appear in your custom form must be added. If you have major changes to make to a form, starting from scratch using Blank Palette might be easier.

Deleting or removing fields on a form is akin to hiding them. The fields are not permanently eliminated and can be added back to the form for future use.

Demo: Customize the Item Entry Master (IC11.1) form
Your instructor will demonstrate how to customize the Item Entry Master (IC11.1) form.
Exercise 3.3: Customize the Item Entry Master (IC11.1) form

In this exercise, you will use the Item Master (IC11.1) form as the base to construct an entry form from a Blank Palette form.

Exercise 3.3 steps

Part 1: Select the Blank Palette form

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select UI Designer.
3. Click OK.
5. Click Next.
6. Select IC (Inventory Control).
7. Click Next.
8. Select IC11.1 (Item Master).
9. Click Next.
10. Confirm that the right form, data area, and system code display on the Confirmation Selections window.
11. Select the Blank Palette radio button.
12. Click Finish. The User Interface Designer window opens with a Blank Palette form.

Part 2: Customize Blank Palette by adding and labeling fields

1. Double-click Label in the UI Toolbox. The Label object is added to the form. Note: You will want to move the object immediately after placing it on the form as all objects are initially placed in the same location on the form.
2. Click Text (…).
3. Type Item Group in the textbox.
4. Click OK. The Label object is renamed Item Group.
5. Double-click Textbox in the UI Toolbox. The Textbox object is added to the form. Note: The textbox is linked to the data source.
6. Select the Textbox object.
7. Drag the Textbox object to align with the Item Group label. Note: Each textbox needs a corresponding label object to identify it to form users.
8. Click Data Src (…).
9. Select ITE-ITEM-GROUP from the list of available data fields.
10. Click OK.
11. Repeat steps 1-9 to add the labels and corresponding textboxes on the form for the following fields.

<table>
<thead>
<tr>
<th>Label</th>
<th>Data Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Name</td>
<td>ITE-ITEM</td>
</tr>
<tr>
<td>Description</td>
<td>ITE-DESCRIPTION</td>
</tr>
<tr>
<td>Description2</td>
<td>ITE-DESCRIPTION2</td>
</tr>
<tr>
<td>Stock Unit of Measure</td>
<td>ITE-STOCK-UOM</td>
</tr>
<tr>
<td>Date Added</td>
<td>ITE-ADDED-DATE</td>
</tr>
</tbody>
</table>

**Part 3: Add default values**

1. Select the **Textbox** object next to the **Item Group** label in the **User Interface Designer** window.
2. Click **Default (…)**.
3. Select the **Literal** radio button.
4. Type **1234** in the **Default Value** field.
5. Click **OK** to return to the **User Interface Designer** window.

**Part 4: Add tooltips and adjust the size of textboxes**

1. Select the **Item Name** textbox in the **User Interface Designer** window.
2. Click **Tooltip (…)**.
3. Type **Enter Item Name** in the textbox.
4. Click **OK**.
5. Select the **Description** textbox.
6. Click **Width**.
7. Type **28** in the **Width** field.
8. Press **Tab**.
9. Click **Tooltip (…)**.
10. Type **Enter Item Description** in the textbox.
11. Click **OK**.
12. Select the **Description2** textbox.
13. Click **Width**.
14. Type **28** in the **Width** field.
15. Press **Tab**.
16. Select the **Stock-Unit of Measure** textbox.
17. Click **Tooltip (…)**.
18. Type Enter Stock Unit of Measure in the textbox.
19. Click OK to return to the User Interface Designer window.

Part 5: Add a line to separate the Required fields from the Additional Information fields
1. Double-click Line in the UI Toolbox.
2. Select the Line object.
3. Drag the Line object to below the Description label to separate the first three items above (Item Group, Item Name, and Description) from the rest of the fields. These are “Required” fields on the form. Adjust the placement of the fields accordingly.
4. Click Width.
5. Type 50 in the Width field.
6. Press Tab.

Part 6: Add buttons for form actions
1. Click the Blank Palette form; do not click in a label or text box. The Name in the Properties pane displays form1.
2. Click Custom (…). The More Document Properties window opens.
3. Click the Actions tab.
4. Select Inquire in the Available Actions section.
5. Click Assign.
6. Select Next.
7. Click Assign.
8. Select Previous.
9. Click Assign.
10. Click Apply.
11. Click OK. The action items are assigned to the form.

Part 7: Preview the form
1. Select View > Preview. The Item Entry Master (IC11.1) preview form opens.
2. Hover the cursor over the Item Name, Description and Stock Unit of Measure fields to view the tooltip.
3. Click X to close the preview form and return to the User Interface Designer window.

Part 8: Save the form
1. Select File > Save As.
2. Type IC11.1_<your initials> in the textbox. Write down the Form Id ______________.
3. Click OK. A custom version of the form is saved.
4. Type ic11.1.<your initials>.xml in the File name field.
5. Click OK. The XML file is saved.
6. Select File > Exit to close the Infor Lawson Design Studio application.
Shortcuts

Often a custom form is created to replace the Infor Lawson-delivered form. If your system is configured to allow users to create shortcuts, you can create and launch custom forms from the shortcut menu of Infor Lawson for Infor Ming.le.

Demo: Create a shortcut

Your instructor will demonstrate how to create a shortcut.

Exercise 3.4: Create a shortcut

In this exercise, you will create a shortcut.

Exercise 3.4 steps

1. Double-click the Infor Lawson for Infor Ming.le (Portal) shortcut on the training desktop. The Infor Lawson for Infor Ming.le login screen opens.
2. Type lawson in the Username field.
3. Type !nfor08 in the Password field.
4. Click Login. The Infor Lawson for Infor Ming.le home page opens.
5. Click the Preferences icon. This icon displays as a cogwheel.
7. Click the Favorites tab.
9. Type My Custom IC11.1 in the textbox.
10. Type LAWFORM|TKN=IC11.1|ID=IC11.1_<your initials> in the textbox to enter the URL. See the IC11.1 Form ID you noted in exercise 3.3, part 8.
11. Click OK.
12. Click Apply.
13. Click OK to return to the Infor Lawson for Infor Ming.le home page Note: After you create the shortcut, the Favorites tab displays on the Infor Lawson for Infor Ming.le (Portal) home page.
14. Click the Favorites tab in the Infor Lawson for Infor Ming.le home page.
Select **My Custom IC11.1** to launch the form.

16. Click **Back** to close **My Custom IC11.1** form and return to the Infor Lawson for Infor Ming.le home page.

**Scenario**
In this scenario, you will create a shortcut for the FR10.1 form you created in exercise 3.2. The syntax to use for the item is:

LAWFORM|TKN=FR10.1|ID=FR10.1_<your initials> in the textbox field. See the FR10.1 Form ID you noted on part 9 of exercise 3.2).

**Demo: Customize the Employee (HR11.1) form functionality for use by Human Resource clerks**
Your instructor will demonstrate how to create a form that will allow Human Resource clerks to quickly look up and change employee addresses.

**Exercise 3.5: Customize the Employee (HR11.1) form functionality for use by Human Resource clerks**
In this exercise, you will create a form that will allow Human Resource clerks to quickly look up and change employee addresses. You need to ensure, however, that the clerk cannot access any other employee data.

**Exercise 3.5 steps**

1. **Part 1: Select Blank Palette to customize the Employee (HR11.1) form**

   1. Log in to **Design Studio** following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application. The **Design Studio Files** window opens.
   2. Select **UI Designer**.
   3. Click **OK**.
   4. Select **APPS10**
   5. Click **Next**.

© 2018 Infor Education. All rights reserved. No part of this document may be reproduced or transmitted in any form, without written permission.
6. Select **HR (Human Resources)**.
7. Click **Next**.
8. Select **HR11.1 (Employee)**.
9. Click **Next**.
10. Confirm the right form, data area, and system code display on the **Confirm Selections** window.
11. Select the **Blank Palette** radio button.
12. Click **Finish**. The **User Interface Designer** window opens with a Blank Palette form.

### Part 2: Add key fields with corresponding labels and textboxes to the Blank Palette form

1. Double-click **Label** in the **UI Toolbox**.
2. Double-click **Textbox** in the **UI Toolbox**.
3. Select the **Textbox** object.
4. Drag the **Textbox** object to align with the **Label** object. **Note:** The order by which the textbox objects are added to the form will determine the tab order.
5. Click **Data Src (…)**.
6. Select **EFFECT-DATE** from the list of data fields. **Hint:** You can use the Find feature to search.
7. Click **OK**.
8. Select the **Label** object.
9. Click **Text (…)**.
10. Type **Effective Date** in the textbox.
11. Click **OK**.
12. Repeat steps 1-11 to add each label and textbox object with corresponding data source to the form.
   - The PRS-NAME textbox can be placed adjacent to EMP-COMPANY and is a display-only field.
   - The EMP-FULL-NAME-CND textbox can be placed adjacent to EMP-EMPLOYEE and is a display only field.

<table>
<thead>
<tr>
<th>Label</th>
<th>Textbox Data Src</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>EMP-COMPANY</td>
</tr>
<tr>
<td>Company Name</td>
<td>PRS-NAME</td>
</tr>
<tr>
<td>Employee</td>
<td>EMP-EMPLOYEE</td>
</tr>
<tr>
<td>Employee Name</td>
<td>EMP-FULL-NAME-CND</td>
</tr>
<tr>
<td>Street Address</td>
<td>EMP-ADDR1</td>
</tr>
<tr>
<td>City</td>
<td>EMP-CITY</td>
</tr>
</tbody>
</table>

Lawson: Customizing Lawson Applications with Design Studio and JavaScript Training Workbook  43

© 2018 Infor Education. All rights reserved. No part of this document may be reproduced or transmitted in any form, without written permission.
Part 3: Set a default value

1. Select the Company textbox.
2. Click Default (…).
3. Select the Literal radio button.
4. Type 4321 in the Default Value field.
5. Click OK to return to the User Interface Designer window.

Part 4: Add buttons for form actions

1. Click the Blank Palette form; do not click in a label or text box.
2. Click Custom (…). The More Document Properties window opens.
3. Click the Actions tab.
4. Select Inquire in the Available Actions section.
5. Click Assign.
6. Select Next.
7. Click Assign.
8. Select Previous.
9. Click Assign.
10. Select Change.
11. Click Assign.
12. Click Apply.
13. Click OK to return to the User Interface Designer window.

Part 5: Add a line object to separate the home and supplemental address

1. Arrange the labels and textboxes Address1, Supplemental City, Supplemental State, and Supplemental Zip Code below the Home address section. Hint: Press and hold the Ctrl key as you click the label and textbox to select multiple objects.
2. Double-click Line in the UI Toolbox.
3. Select the Line object.
4. Drag the Line object to separate the top and bottom sections of the form.
5. Click Width.
6. Type 38 in the Width field.
7. Click Text (…).
8. Type Supplemental Address in the textbox.
9. Click OK to return to the User Interface Designer window.

Part 6: Add tooltips
1. Select the Employee textbox in the User Interface Designer window.
2. Click Tooltip (…).
3. Type Enter Employee Number in the textbox field.
4. Click OK.

Part 7: Preview the form
2. Click Next to view data on the form. Note: The Next icon is a right-facing arrow.
3. Click X to close the preview window and return to the User Interface Designer window.

Part 8: Save the form
1. Select File > Save As.
2. Type HR11.1__<your initials> in the textbox. Write down the Form Id _____________.
3. Click OK. A custom form is saved.
4. Type hr11.1.<your initials>.xml in the File name field. The XML file is saved.
5. Click OK.
6. Select File > Exit to close the Infor Lawson Design Studio application.
Scenario

In this scenario, you will customize the Employee form (HR11.1) for self-service. You will also create a shortcut for the new form on the Infor Lawson for Infor Ming.le (Portal) site.

- Create labels and textboxes with corresponding data source for the table below. Add default values where identified.

<table>
<thead>
<tr>
<th>Label</th>
<th>Textbox Data Src</th>
<th>Default</th>
<th>Output Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>EMP-COMPANY</td>
<td>4321</td>
<td>X</td>
</tr>
<tr>
<td>Company Name</td>
<td>PRS-NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Employee Number</td>
<td>EMP-EMPLOYEE</td>
<td>1000</td>
<td>X</td>
</tr>
<tr>
<td>Employee Name</td>
<td>EMP-FULL-NAME-CND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street Address</td>
<td>EMP-ADDR1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>EMP-CITY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>EMP-STATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zip Code</td>
<td>EMP-ZIP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Create a button and set the initial form action to Inquire.
- Preview the form.
- Close the preview window.
- Save the form as HR11.1A_<your initials>.
- Save the XML file as HR11.1A_<your initials>.xml
- Create a shortcut for the form on the Infor Lawson for Infor Ming.le (Portal) site. Use the following value: LAWFORM|TKN=HR11.1|ID=HR.11.1A_XX

Note: The default values of 4321 and 1000 would be used for testing only. Once testing is complete, the default value for EMP-COMPANY would be changed to UserEnv > company and the default value for EMP-EMPLOYEE would be changed to UserEnv > employee.
Check your understanding

Which of the following are examples of form customizations using UI Designer? Select all that apply.

a) Changing the tabbing order on a form  
b) Removing tabs from a form with multiple tabs  
c) Removing fields  
d) Adding images or logos to a form  
e) Exposing hidden fields

When using __________, the form appears in the design window with all the fields and other objects from the original Infor Lawson form in place.

a) Blank Palette  
b) QuickPaint  
c) Portal

True or False: You are allowed to overwrite the Infor Lawson version of the form.

a) True  
b) False

True or False: When you are significantly modifying an Infor Lawson form (for example, removing or rearranging many of the fields), starting from a Blank Palette might be easier than QuickPaint.

a) True  
b) False

In __________, the toolbox includes objects such as radio buttons and text input boxes that allow users to select or add data to the form.

a) Wizard Designer  
b) Custom Page Designer  
c) UI Designer

__________ are aspects or values of the object that you can change such as text, font color, width size, and default value.

a) Properties  
b) Toolboxes  
c) Objects
The order by which the _________ objects are added to the form will determine the tab order.

a) Label
b) Textbox
c) Line

Which form custom property option allows users to modify whether an end user can Inquire, Add, or Change a form?

a) Transfers
b) General
c) Actions
d) Data

The section in the bottom of the form with Design/Source/Script is the______________.

a) Navigation bar
b) View menu bar
c) Browser
d) Transfer bar

This object type is typically used adjacent to a textbox so users can identify the field on the form.

a) Rectangle
b) Image
c) Label

Adding this type of property to a textbox/field enables forms to limit data to a specific employee, requester, etc. The form can then be used as a self-service form.

a) Default value
b) Scrolling
c) Border

An object that users can click to place a check mark and toggle between two settings.

a) Text area
b) Checkbox
c) Detail area
Lesson 4: Custom Page Designer

Estimated time
6 hours

Learning objectives
After completing this lesson, you will be able to:

- Identify types of objects and their uses within Custom Page Designer.
- Describe the process for building a data query object and custom menu.
- Define the steps to map both data and non-data objects.

Topics
- Custom Page Designer overview
- Custom Page Designer objects
  - Linked objects
  - Menu objects
  - Special objects
- Querying Infor Lawson data
- Mapping data query objects
- Check your understanding
Custom Page Designer overview

A custom page is a front-end or entry point to Lawson (as well as other locations). When you create a custom page, you are creating a unique interface to Lawson. In other words, you are designing how Lawson looks and feels to a user.

You can use the Custom Page Designer to:

- Create custom pages
- Control how an object appears on the page
- Provide an entry point to any location (not just Infor Lawson), such as an Internet site or your company's intranet
- Provide entry points to Infor Lawson reports
- Create custom menus
- Create queries to access Infor Lawson data using:
  - Data queries
  - Transaction queries
  - Drill servlets
- Add images or custom colors
- Add custom messages

Custom Page Designer design window

The design window is intended to reflect a custom page with navigation items on the left pane and content items on the right. The Toolbox is on the left-side pane and the Properties pane is on the right-side of the screen.

The Custom Page Designer design window has two toolboxes, Lawson and General.

- The Lawson toolbox is for adding Infor Lawson objects, including Infor Lawson menus, reports, and objects for retrieving Infor Lawson data to a custom page.
- The General toolbox is for adding objects that might not be related to Infor Lawson, including RSS web sites, images, and user-defined menus.

Customizations to Infor Lawson Design Studio pages are stored in the LAWDIR/persistdata/lawson/portal/content/pages directory. Each object that is added to the page is stored in subdirectories according to its type.
Custom Page Designer objects

You can control the objects (navigation and content items) that appear in the custom page, where they appear, and, in many respects, how they behave.

In Custom Page Designer, many of the available objects are related to retrieving data, rearranging menus, and allowing users to navigate to URLs from a custom page.

When you choose to create a new custom page, the Custom Page Designer design window is always blank. All objects must be added.

Linked objects

Linked objects are objects that link a custom page to a web site, report, or an image. The objects display directly on the page and include the following properties:

<table>
<thead>
<tr>
<th>Objects</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Add a link to a .gif or .jpg image file to a custom page as well as set custom properties to control the appearance of the image on the page</td>
</tr>
<tr>
<td>Report</td>
<td>Add a link to the Infor Lawson report utility directly to the custom page</td>
</tr>
<tr>
<td>Rich Site Summary (RSS) news</td>
<td>Add a link to a Rich Site Summary news site into a custom page to receive What's New type of headlines</td>
</tr>
<tr>
<td>Web page</td>
<td>Add a link to any external web site such as Google, Weather, or your own company's site</td>
</tr>
</tbody>
</table>

Demo: Create an image, text, and a Web page

Your instructor will demonstrate how to create an image, text, and a Web page.

Exercise 4.1: Create an image, text, and a Web page

In this exercise, you will create an image, text, and a Web page.
Exercise 4.1 steps

Part 1: Add an Infor logo image

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select Custom Page Designer.
3. Click OK. The Custom Page Designer window opens.
4. Click the Design tab if not already open.
5. Select General from the Toolbox. Note: If the General toolbox is not active, click the General button that appears at the bottom of the Toolbox.
7. Type 100 for both Height and Width in the Properties pane. Note: Remember to press Tab to move from field to field in the Properties pane, otherwise your entry may not save.
8. Click Custom (...). The Property Page window opens.
10. Click OK to return to the Custom Page Designer window.

Part 2: Add a text object

1. Double-click Text in the General toolbox. A Text object is added to the main Custom Page Designer window.
2. Type or select the property values specified in this table for the Text object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td>258</td>
</tr>
<tr>
<td>Top</td>
<td>100</td>
</tr>
<tr>
<td>Height</td>
<td>50</td>
</tr>
<tr>
<td>Width</td>
<td>200</td>
</tr>
<tr>
<td>Border</td>
<td>No</td>
</tr>
<tr>
<td>Scroll</td>
<td>Yes</td>
</tr>
</tbody>
</table>

3. Click Custom (...). The Text Property Page opens.
4. Type Infor Website in the text field. Note: You can also modify the properties of the text such as font type, size, color, etc.
5. Click Apply.
6. Click OK to return to the Custom Page Designer window.
Part 3: Add a Web page


2. Type or select the property values specified in this table for the Web Page object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td>48</td>
</tr>
<tr>
<td>Top</td>
<td>224</td>
</tr>
<tr>
<td>Height</td>
<td>600</td>
</tr>
<tr>
<td>Width</td>
<td>600</td>
</tr>
<tr>
<td>Border</td>
<td>No</td>
</tr>
<tr>
<td>Scroll</td>
<td>Yes</td>
</tr>
<tr>
<td>Auto hide</td>
<td>No</td>
</tr>
</tbody>
</table>

3. Click Custom (…). The Property Page dialog window opens.

4. Type http://www.infor.com in the Address field.

5. Click OK to return to the Custom Page Designer window.

Part 4: Preview and save file


2. Click X to close the preview window.

3. Select File > Save As. Note: If prompted, select Remote instead of Local.

4. Type myinforpage in the File name field. Note: You will be prompted to save each object on the custom page as an xml file.

5. Click OK.

6. Type imagelogo1 in the File name field.

7. Click OK. The image object is saved.

8. Type textinfor1 in the File name field.

9. Click OK. The text object is saved.

10. Type htmfinforwebpage in the File name field.

11. Click OK. The Web page object is saved.

12. Select File > Close to close the file.
Shortcut
As with forms, you can create a shortcut for your custom page using the Infor Lawson for Infor Ming.le shortcut menu.

Demo: Create a custom page shortcut
Your instructor will demonstrate how to create a custom page shortcut.

Exercise 4.2: Create a custom page shortcut
In this exercise, you will create a custom page shortcut.

Exercise 4.2 steps
1. Log in to Infor Lawson for Infor Ming.le (Portal) following the steps 1-4 in exercise 3.4.
2. Click Preferences.
3. Select User Options.
4. Click the Favorites tab.
5. Click New.
6. Type My Custom Infor Page in the Item Name text box.
7. Type LAWPAGE_file=myinforpage.xml in the Item URL text box.
8. Click OK.
9. Click Apply.
10. Click OK.
11. Click the Favorites tab in the Infor Lawson for Infor Ming.le home page. Note: The custom page now appears in the Favorites list along with the custom forms you created in earlier exercises.
12. Select My Custom Infor Page to view the custom page.
13. Click X to close the My Custom Infor Page.
14. Click X to close Infor Lawson for Infor Ming.le.
**Reports**

When a report object is added to a user's custom page, the user's jobs are immediately available any time the custom page is opened. Options for submit, print, and delete can be added to a report object.

---

**Demo: Create a report object**

Your instructor will demonstrate how to create a report.

**Demo steps**

1. Double-click the **Design Studio** shortcut on the training desktop.
2. Click **New**.
3. Select **Custom Page Designer**.
4. Click **OK**.
6. Type or select the property values in the table below for the **Report** object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td>100</td>
</tr>
<tr>
<td>Top</td>
<td>100</td>
</tr>
<tr>
<td>Height</td>
<td>400</td>
</tr>
<tr>
<td>Width</td>
<td>400</td>
</tr>
<tr>
<td>Border</td>
<td>Yes</td>
</tr>
<tr>
<td>Scroll</td>
<td>Yes</td>
</tr>
<tr>
<td>Print</td>
<td>Yes</td>
</tr>
<tr>
<td>Submit</td>
<td>No</td>
</tr>
<tr>
<td>Delete</td>
<td>Yes</td>
</tr>
</tbody>
</table>

7. Select **View > Preview**. The **Custom Page** preview window opens displaying the report. The report allows the user to Print and Delete a report but not Submit.
8. Click **X** to close the preview window.
9. Change the **Submit** property to **Yes** to allow a user to submit a report.
10. Select **View > Preview**. The user is now allowed to submit a report.
11. Click **X** to close the preview window.
12. Select **File > Save As**. If prompted, select Remote.
13. Type *myreport* in the **File name** field.
14. Click **OK**. The custom report is saved.
15. Type *report1* in the **File name** field. The **XML file** is saved.
16. Click **OK** to return to the **Custom Page Designer** window

---

**Menu objects**

Menu objects are objects that add navigation items to a custom page. You can add a predefined Infor Lawson menu or a custom menu.

<table>
<thead>
<tr>
<th>Objects</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lawson Menu</td>
<td>Add a list of Infor Lawson forms or pre-defined menus to the navigation pane; users can select an item in the list to access an Infor Lawson form</td>
</tr>
<tr>
<td>Local Transfer Menu</td>
<td>Add a list based on an existing local transfer menu for an Infor Lawson form</td>
</tr>
<tr>
<td>User Defined Menu</td>
<td>Add links that you define to the navigation pane of a custom page</td>
</tr>
</tbody>
</table>

---

**Demo: Build a custom menu**

Your instructor will demonstrate how to build a custom menu.
Exercise 4.3: Build a custom menu

In this exercise, you will build a customized menu for the HR forms most frequently used by HR Clerks.

Exercise 4.3 steps

Part 1: Create the hrm_menu

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select Custom Page Designer.
3. Click OK.
4. Select Lawson in the Toolbox.
5. Double-click Lawson Menu in the Lawson toolbox. The Lawson Menu is added to the Custom Page Designer.
6. Click Custom (...). The Property Pages window opens.
7. Select APPS10 in the Data Area field. Note: The data area, system, and Lawson program are selected from a drop-down list.
8. Select HR Human Resources in the System field.
9. Select HRMN.0 Human Resources in the Lawson Program field. The Available Menus/Programs column displays the HRMN.0 options.
10. Select HRMN.1 Setup.
11. Click the right-facing arrow to send the item from the Available Menus/Programs list to the Selected Menus/Programs list.
12. Select HRMN.2 Processing.
13. Click the right-facing arrow.
14. Click Apply.
15. Click OK.
16. Type hrm_menu in the Name field in the Properties pane.
17. Press Tab.
18. Type HR Menu in the Title field.
19. Press Tab.
20. Click Navigate in the Properties pane.
21. Select Replace parent from the available options in the Navigate field.

Part 2: Preview and save the custom menu

2. Select Custom Page > HR Menu > HRMN.2 Processing. The HR Processing forms display.
3. Click X to close the preview window.
4. Select File > Save As. Note: Select Remote instead of Local if an error box appears.
5. Type mycustompage1 in the File name field. Note: The file type will be *.htm, or *.html.
6. Click **Save**.
7. Click **OK** to save hrm_menu.xml file.

**Part 3: Create a user-defined menu**
1. Select the **General** toolbox menu.
2. Double-click **User Defined Menu**.
3. Type **my_menu** in the **Name** field.
4. Type **Websites** in the **Title** field.
5. Click **Navigate**.
6. Select **New Window** in the **Navigate** field.
7. Click **Custom (…)**.
8. Select **new** from the **Available Menus/Programs** panel.
9. Click the right-facing arrow to send the item to the **Selected Menus/Programs** panel.
10. Type **Weather** in the **Caption** field.
11. Type **Weather** in the **Description** field.
12. Type **http://www.weather.com** in the **URL** field.
13. Click **Apply**.
14. Click **OK**.

**Part 4: Preview and save**
1. Select **View > Preview**. The **Custom Page** preview window opens.
3. Click **X** to close the Web page.
4. Click **X** to close the preview window.
5. Select **File > Save**. You will be prompted to save the new objects on the custom page.
6. Click **OK** to save.
   
   Do not close the Custom Page Designer as you will use it for the next exercise.
**Special objects**

Special objects such as text, composite, and formlets have special purposes in the Custom Page Designer.

<table>
<thead>
<tr>
<th>Objects</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Add explanatory text to a custom page</td>
</tr>
<tr>
<td>Composite</td>
<td>Insert a group of objects such as a report, data query, or an image – already saved as a custom page - into an existing custom page</td>
</tr>
<tr>
<td>Formlet</td>
<td>Add a previously defined form into a custom page</td>
</tr>
</tbody>
</table>
Querying Infor Lawson data

Data retrieval objects let you make queries to Infor Lawson data and retrieve the data in a custom page. The Infor Lawson Design Studio includes the API Builder feature that allows a user to retrieve and send Infor Lawson data through an Internet Object Service (IOS) engine. The IOS acts as a conduit between Infor Lawson’s web products and the Infor Lawson database. The three main IOS engines are:

1. **Transaction Service**

   The Transaction Service allows users to perform application transactions in the Infor Lawson system via the web. A Transaction Service utilizes the defined business logic to perform transactions. Example user requests include:
   - Add/Change
   - Inquiry
   - Delete
   - Next/Previous

2. **Data Service**

   The Data Service allows users to perform read-only queries on the Infor Lawson system via the web server. A Data Service call dynamically selects records from an Infor Lawson database. Example user requests include:
   - Query strings (data queries) against database tables

3. **Drill Service**

   The Drill Service allows users to perform read only questions to Infor Lawson Drill Around data via the web server. A Drill Service uses predefined Drill Around screen rules and object rules when users request a Drill Around or Select list.

**Data query custom properties**

This Data Query Property Pages dialog box provides many features for building the data query and for formatting the resulting data. The dialog box is organized into tabs:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>This tab contains the API Builder (click the button to open a dialog box of options). The API is the actual query string that you will build.</td>
</tr>
<tr>
<td>Field</td>
<td>This tab contains the Header and Body tabs. This option is used for formatting the results of the query.</td>
</tr>
<tr>
<td>Index</td>
<td>This tab is used to make selections for index values.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Value</strong> – Select a default value for the key field.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Literal</strong>: The default value is a literal string.</td>
</tr>
<tr>
<td></td>
<td>- <strong>UserEnv</strong>: The default value will be a value from the environment file for the user.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Trigger Refresh</strong> - When objects are mapped, any change to a custom page data source will refresh the object with new data.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Hide</strong> – If a filter is used, the data from this key will not appear in the filter.</td>
</tr>
<tr>
<td>Condition</td>
<td>This tab is used to specify selection conditions for the query.</td>
</tr>
</tbody>
</table>

© 2018 Infor Education. All rights reserved. No part of this document may be reproduced or transmitted in any form, without written permission.
<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria</td>
<td>This tab is used in the rare case when a string search is the only way to retrieve the data. It is not recommended for large volumes of data, as it can make the object slow to load, whereas indexes are more efficient and should be used.</td>
</tr>
</tbody>
</table>

**Demo: Build a data query object**

Your instructor will demonstrate how to build a data query object.

**Exercise 4.4: Build a data query object**

In this exercise, you will build a custom page that provides different pieces of information for HR Clerks: employee name, employee status, pay rate, and hire date.

**Exercise 4.4 steps**

**Part 1: Create a data query object**

1. Select Lawson in the Toolbox. **Note:** You will use the same MyCustomPage1.xml from exercise 4.3.
2. Double-click the Data Query object. The Data Query object is added to the main Custom Page Designer window.
3. Type or select the property values specified in this table for the Data Query object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>empdata</td>
</tr>
<tr>
<td>Left</td>
<td>0</td>
</tr>
<tr>
<td>Top</td>
<td>50</td>
</tr>
<tr>
<td>Height</td>
<td>200</td>
</tr>
</tbody>
</table>
4. Click Custom (…). The **Data Query Property Pages** window opens.

5. Click **API (…)**. The **API Builder** window opens.

6. Type or select the data specified in the table below to create the query:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Lines</td>
<td>APPS10</td>
</tr>
<tr>
<td>System Code</td>
<td>HR Human Resources</td>
</tr>
<tr>
<td>Data Tables</td>
<td>EMPLOYEE</td>
</tr>
</tbody>
</table>
| Table Elements        | COMPANY
                        EMPLOYEE
                        LAST-NAME
                        FIRST-NAME
                        EMP-STATUS
                        DATE-HIRED
                        PAY-RATE |
| Indices               | EMPSET1                |

7. Click **Build Query** to create the data query.

8. Click **OK** to return to the **Data Query Property Pages** window.

9. Click the **Index** tab.

10. Type 4321 in the **Company Value** field.

11. Type 1000->1010 in the **Employee Value** field. **Note:** Defining a value using a hyphen and greater than symbols indicates a range of values within the specified criteria. In this example, employees with employee numbers ranging from 1000 to 1010 will be included.

12. Click the **Fields** tab.
13. Click the **Body** tab.
14. Select the **Hide Field** checkbox for the **Company** field.
15. Click **Apply**.
16. Click **OK** to save your query settings and return to the **Custom Page Designer** window.

**Part 2: Preview and save**

1. Select **View > Preview** to test your data. The **Custom Page** preview window opens.
2. Click the **Employee** field to view the list of employees.
3. Click **X** to close the preview window.
4. Select **File > Save**. This prompts you to save the new objects you’ve added to the custom page.
5. Click **OK**. The **custom page** is saved.
6. Click **OK** to save **empdata.xml** file. The **XML file** is saved.

⚠️ When entering parameters in the Properties pane, use the Tab key to move from one property to another. Otherwise, your input may not “take.”
Mapping data query objects

Data mapping allows different custom page objects to share data. A common use for mapping data is to link two objects so that they will display related data simultaneously on the custom page. For example, you may want to link an employee image with corresponding employee data so that when a user views the custom page and clicks the employee data—such as an employee number—the corresponding employee image displays.

Data mapping is also used to map a variable for a custom page object that stores the information the custom page sends when the user initiates an action with that object. For example, a URL to a Web Page object can include an embedded variable that will allow the Web Page object to display a very specific web page based on the related data.

Demo: Map data query objects

Your instructor will demonstrate how to map a data query object.

Exercise 4.5: Map data query objects

In this exercise, you will build on exercise 4.4 by mapping a data query object which allows different portal objects to share data.

Exercise 4.5 steps

Part 1: Add a second data query object to the custom page

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Click the Existing tab.
4. Click OK.
5. Select your previously saved custom page MyCustomPage1.xml from the list of folders/files.
6. Click OK.
7. Double-click Data Query in the Lawson toolbox menu.
8. Type or select the property values specified in this table for the data query object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>autodata</td>
</tr>
<tr>
<td>Left</td>
<td>0</td>
</tr>
<tr>
<td>Top</td>
<td>263</td>
</tr>
<tr>
<td>Height</td>
<td>200</td>
</tr>
<tr>
<td>Width</td>
<td>200</td>
</tr>
<tr>
<td>Border</td>
<td>No</td>
</tr>
<tr>
<td>Scroll</td>
<td>Yes</td>
</tr>
<tr>
<td>Filter</td>
<td>No</td>
</tr>
<tr>
<td>Filter Title</td>
<td></td>
</tr>
<tr>
<td>Visibility</td>
<td>Visible</td>
</tr>
<tr>
<td>Navigate</td>
<td>Pass to Parent</td>
</tr>
</tbody>
</table>

9. Click **Custom (…)**.

10. Click **API (…)**.

11. Type or select the property values specified in this table:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Lines</td>
<td>APPS10</td>
</tr>
<tr>
<td>System Code</td>
<td>PA Personnel</td>
</tr>
<tr>
<td>Data Tables</td>
<td>EMPAUTO</td>
</tr>
<tr>
<td>Table Elements</td>
<td>COMPANY, EMPLOYEE, MODEL, MAKE, YEAR, LIC NBR, LIC STATE</td>
</tr>
</tbody>
</table>
Part 2: Save the data query object
1. Select File > Save.
2. Click OK. The custom page is saved.
3. Click OK to save the autodata.xml file. The XML file is saved.

Before setting the Auto hide property to Yes, be sure to complete customizing the data query object, otherwise, you will not be able to see the results.

Part 3: Add a third data query object to the custom page
1. Double-click Data Query in the Lawson toolbox menu. A third data query object is added to the main Custom Page Designer window.
2. Select the Data Query object.
3. Drag the Data Query object so it is to the right of the autodata object.
4. Type or select the following property values in this table for the data query object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>benda</td>
</tr>
<tr>
<td>Left</td>
<td>219</td>
</tr>
<tr>
<td>Top</td>
<td>266</td>
</tr>
</tbody>
</table>

Lesson 4: Custom Page Designer
5. Click Custom (…).
6. Click API (…).
7. Type or select the property values specified in this table on the API Builder window:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>200</td>
</tr>
<tr>
<td>Width</td>
<td>300</td>
</tr>
<tr>
<td>Border</td>
<td>No</td>
</tr>
<tr>
<td>Scroll</td>
<td>Yes</td>
</tr>
<tr>
<td>Filter</td>
<td>No</td>
</tr>
<tr>
<td>Filter Title</td>
<td></td>
</tr>
<tr>
<td>Visibility</td>
<td>Visible</td>
</tr>
<tr>
<td>Navigate</td>
<td>Pass to Parent</td>
</tr>
</tbody>
</table>

8. Click Build Query.
9. Click OK to return to the Data Query Properties page.
10. Click the Fields tab.
11. Click the Header tab.
12. Type *Yearly Contribution* in the EMP-AFT-CONT Caption field. This helps the user better identify the field on the custom page.

13. Click the Body tab.

14. Select the Hide Field checkboxes for Company and Employee to hide both fields on the custom page.

15. Click the Index tab.

16. Type 4321 in the COMPANY Value field.

17. Type RS in the PLAN-TYPE Value field.

18. Select the Trigger Refresh checkboxes for Company and Employee.

19. Click Apply.

20. Click OK to return to the Custom Page Designer window.

21. Select Yes in the Auto hide property in the Properties pane for the bendata object.

**Part 4: Map the three data queries.** For this part, you will map the COMPANY and EMPLOYEE field from the empdata object to the page, then map the COMPANY and EMPLOYEE fields from the other queries.

1. Click a whitespace in the Custom Page Designer window. **Hint:** Place the cursor above the empdata object and confirm that the Properties pane Name displays as portalpage.

2. Click Custom (...). The Data Map window opens.

3. Select empdata from the Objects drop-down list. The left panel populates the fields selected for the empdata object.

4. Select COMPANY from the left panel and click the right-facing arrow. COMPANY moves to the right panel.

5. Select EMPLOYEE from the left panel and click the right-facing arrow. EMPLOYEE moves to the right panel.

6. Select autodata from the Objects drop-down list. The left panel displays the fields for autodata.

7. Select COMPANY from the left panel.

8. Select COMPANY from the right panel.

9. Click Map. The Mapped elements items change.

10. Select EMPLOYEE from the left panel.

11. Select EMPLOYEE from the right panel.

12. Click Map.

13. Select bendata from the Objects drop-down list.

14. Select COMPANY from the left panel.

15. Select COMPANY from the right panel.

16. Click Map.

17. Select EMPLOYEE from the left panel.

18. Select EMPLOYEE from the right panel.

19. Click Map.

20. Click Apply.

Lesson 4: Custom Page Designer
21. Click **OK** to return to the **Custom Page Designer** window.

**Part 5: Preview and save**

1. Select **File > Save**. The application will prompt you for the objects that have yet to be saved at this point in the exercise.
2. Click **OK** several times to save the file objects.
3. Select **View > Preview**. The **Custom Page** preview window opens. **Note:** Because you have set the second and third data queries to Auto Hide, you will not see them until you click on a record from the first data query.
4. Click **Employee 1000**.
5. Click **Employee 1003**. The **empdata** triggers the data for the other information to change.
6. Click **X** to close the preview window.
7. Click **X** to close the **MyCustomPage1.xml** file.

**Demo: Map non-data objects**

Your instructor will demonstrate how to map a non-data object.

---

**Exercise 4.6: Map non-data objects**

In this exercise, you will add a box for an employee picture on the custom page containing the mapped Employee, Auto and Benefits data queries. You will want the image to be based on the specific employee record being selected. All employee images are stored online and each image is named as an employee number.

**Exercise 4.6 steps**

**Part 1: Create employee image object**

1. Log in to **Design Studio** following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Click the **Existing** tab.
4. Click **OK**.
5. Select your previously saved custom page **MyCustomPage1.xml** from the list of folders/files.
6. Click **OK**.
7. Press and hold down the **CTRL** key, then select the **empdata, autodata, and bendata** objects.
8. Press the **down arrow** several times to make room for two images on the top of the custom page.
9. Select the **General** toolbox.
10. Double-click **Image**. The **Image** object is added to the page.
11. Type or select the property values specified in this table for the image object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>emppic</td>
</tr>
<tr>
<td>Left</td>
<td>52</td>
</tr>
<tr>
<td>Top</td>
<td>8</td>
</tr>
<tr>
<td>Height</td>
<td>100</td>
</tr>
<tr>
<td>Width</td>
<td>100</td>
</tr>
</tbody>
</table>

12. Click **Custom (…)**.
13. Type **emp** in the **Name** field. **Note**: Leave the Value field blank.
14. Click **Add**.
15. Type **http://lsf10.gdeinfor2.com/lawson/images/<<emp>>.jpg** in the **Address** field.
16. Click **OK** to close the window and return to the **Custom Page Designer** window.

**Part 2: Map the emppic object to EMPLOYEE**

1. Click anywhere that is a white space on the design window. The **Properties** will display as **portalpage**.
2. Click **Custom (…)**.
3. Select **emppic** in the **Objects** menu.
4. Select **emp** in the left panel.
5. Select **EMPLOYEE** in the right panel.
6. Click **Map**.
7. Click **Apply**.
8. Click **OK**.

**Part 3: Preview the custom page**

1. Select **View > Preview**. The **Custom Page Designer** preview window opens.
2. Select **Employee 1000**.
3. Select **Employee 1001**. The employee image changes with each selection.
4. Click **X** to close the preview window and return to the **Custom Page Designer** window.
**Part 4: Add a logo image to the custom page**

1. Select the **General** toolbox.
2. Double-click **Image**.
3. Type or select the property values specified in this table for the **logo** object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>logo</td>
</tr>
<tr>
<td>Left</td>
<td>450</td>
</tr>
<tr>
<td>Top</td>
<td>8</td>
</tr>
<tr>
<td>Height</td>
<td>100</td>
</tr>
<tr>
<td>Width</td>
<td>100</td>
</tr>
</tbody>
</table>

4. Click **Custom (...)**.
6. Click **OK** to close the window and return to the **Custom Page Designer** window.

**Part 5: Preview and save the custom page**

1. Select **View > Preview**. The **Custom Page Designer** preview page opens. The **Infor logo** appears on the custom page.
2. Click **X** to close the preview window.
3. Select **File > Save**. You will be prompted to save the image objects for the custom page.
4. Click **OK** to save each image object.
Check your understanding

__________ objects are those that link a custom page to a web site, report, or an image.
   a) Formatted
   b) Linked
   c) Special

Which of the following are linked objects? Select all that apply.
   a) Web page
   b) Image
   c) Formlet
   d) Report

__________ objects add navigation items to a custom page.
   a) Menu
   b) Linked
   c) Special

Which of the following are examples of Special objects in Custom Page Designer? Select all that apply.
   a) Formlet
   b) Text
   c) Composite
   d) Mapping

__________ is an API string that calls the database directly.
   a) Transaction Service
   b) Drill Service
   c) Data Service
objects let you make queries to Infor Lawson data and retrieve the data in a custom page.

a) Data retrieval
b) Label
c) Lawson menu

Which tab would you select if you want to enter a literal default value string for a field?

a) General
b) Condition
c) Index

Which Index tab option would you select if you have two mapped objects and you want any change to a custom page data source to refresh the object with new data?

a) Hide
b) Trigger refresh
c) Value
Lesson 5: Enhancing Infor Lawson Design Studio with JavaScript

Estimated time
7 hours

Learning objectives
After completing this lesson, you will be able to:
- Define how to extend functionality of a form using JavaScript.
- Describe the steps to designate a required field.
- Define ways to limit the options available in a user’s selection list.
- Describe how to combine multiple functions using JavaScript.
- List the steps to perform a calculation on a form that’s been modified with a new field.
- Explain the process for using Custom Page Designer to create a custom page for approving invoices.

Topics
- JavaScript overview
- Custom scripting in Infor Lawson Design Studio
- Working with multiple forms
- Enhancing a custom page with JavaScript
- Check your understanding
JavaScript overview

Infor Lawson Design Studio User Guide
O'Reilly series (http://www.oreilly.com) on Web design/development
Lawson Knowledge Base
JavaScripting resources: http://www.w3schools.com

What is JavaScript?
JavaScript is a scripting language developed by Netscape to enable web authors to design interactive sites. Although it shares many features and structures of the full Java language, it was developed independently. JavaScript can interact with Hypertext Markup Language (HTML) source code enabling web authors to add dynamic content to their sites.

Helpful IT industry terms
Extensible Markup Language (XML) – XML allows designers to create their own customized tags, enabling the definition, transmission, validation and interpretation of data between applications and between organizations.

XML components:
- Document Type Definitions (DTD): The DTD defines the structure of the document.
- Cascading Style Sheets (CSS): CSS defines how elements appear.

XML vs. HTML – Both XML and HTML are markup languages. When you inquire with both HTML and XML, data is returned through the browser. With HTML you have a predefined set of tags that you must use. With XML, you define the tags that you want to use. XML does not replace HTML. It can be used to enhance HTML. XML documents can be linked to HTML and HTML can be translated to XML.

Extensible Style Sheets (XLS) – XLS dictates the way a web page is viewed or printed.

Data Island – Data Island is the XML data source maintained behind the HTML pages.
Custom scripting in Infor Lawson Design Studio

JavaScript programming extends the functionality of the Infor Lawson Design Studio. The Infor Lawson Design Studio includes a development environment and some tools for working with scripting.

- Use the script screen and Design Studio’s built in functions
- Customize the Infor Lawson Design Studio by accessing and using scripting that is not available in the interface

UI Designer

UI Designer (without custom script) can only alter existing properties of forms, it cannot combine the two forms (because doing so would involve adding functionality to one of them). With custom scripting you can solve this issue.

Custom Page Designer

You can use JavaScript in adding interactivity to custom pages that have been created with Custom Page Designer.

UI Designer scripting environment

The main scripting window includes a text editing tool that provides auto-indenting. The right pane of the window displays data elements and functions that are available for a form.

Objects

When you make certain selections within the environment, a JavaScript function stub appears automatically in the script window.

Events

An event is an action that you can associate with an object. When an object is selected, a list of events available to be used with the object appears in the Events drop-down box. When you select the event, a JavaScript function stub for that event and object appears automatically in the script window.

Functions/Objects

Function view - shows the methods that are available via the form object
Data Objects view - shows the transaction fields (but not field names)

Note: All data objects, including those that are hidden on the form are available.

Include Files

Include Files provide a way for developers to group functions so they can be re-used within customizations. For example, the FORM_OnInit event can be used in conjunction with a custom function the developer creates (i.e. initCompanyField) to disable input and use a default value for the Company field on a form.
Object Viewer

Object Viewer is a tool that provides reference material about Infor Lawson objects and methods that can be accessed through the scripting interface. It is available from both UI Designer and Custom Page Designer’s Tools menu.

Object Viewer sections:
- Lawson for Infor Ming.le – contains objects related to Infor Lawson for Infor Ming.le
- Form – contains objects that interact with Lawson forms
- Pages – contains objects related to custom pages created in Custom Page Designer

Demo: Accessing Infor Lawson Design Studio’s Object Viewer

Your instructor will demonstrate how to access the Object Viewer.

Demo steps

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select Custom Page Designer.
3. Click OK.
5. Click the plus (+) sign to expand the Forms folder from the left panel.
6. Select LawFormObj > Methods. A list of all of the methods for that object displays.
7. Select getDataValue. The getDataValue parameters, remarks, and an example of how the method may be used within JavaScript displays.
8. Click the plus (+) sign to expand the Custom Pages folder from the left panel.
10. Select getValue. The getValue parameters, remarks, and an example of how the method may be used within JavaScript displays.
11. Click X to close the Lawson Portal Object Viewer and return to the Custom Page Designer window.
Lesson 5: Enhancing Infor Lawson Design Studio with JavaScript

Viewing the methods and parameters is a useful reference for Scripting within Infor Lawson Design Studio. In addition to providing the methods available for objects, examples and remarks are available.

Demo: Add to the Quick Item Entry Form Functionality with JavaScript

Your instructor will demonstrate how to add to the Quick Item Entry Form functionality with JavaScript.

Exercise 5.1: Add to the Quick Item Entry Form Functionality with JavaScript

In this exercise, you will add to the Quick Item Entry Form functionality with JavaScript.

Exercise 5.1 steps

Part 1: Add a vendor’s image to the IC11.1 form

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select File > Open.
3. Select UI Designer.
4. Click OK. The Open Custom User Interface dialog window opens.
5. Select APPS10 in the Data area field.
6. Click Find.
7. Select IC11.1_<your initials>.xml from exercise 3.3 in the list of forms displayed.
8. Click Open. The User Interface Designer window opens.
9. Double-click Image in the UI Toolbox. The Image object is added to the form.
10. Type or select the property values specified in this table for the image.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>grainger</td>
</tr>
<tr>
<td>Top</td>
<td>2</td>
</tr>
</tbody>
</table>
11. Click **Source (…)** in the **Properties** pane.
12. Select **logo_grainger.jpg** from the **Image List**.
13. Click **OK**. This adds the vendor image to the form.

The Script tab is the development window for creating scripts using JavaScript in Infor Lawson Design Studio. The code, function, methods, and events are either manually typed in this white space or are automatically enabled depending on user preference.

**Part 2: Add an image OnClick function**

1. Click the **Script** tab. The scripting window displays. **Notes:**
   - The **Script** tab is on the bottom of the **User Interface Designer** window.
   - The **Objects** and **Events** menus, as well as the **Functions** and **Elements** lists, are located in the right side of the **User Interface Designer** scripting window.
2. Select **Image** from the **Objects** menu.
3. Select **OnClick** from the **Events** menu. This places the function and the base tags around the function on the **UI Interface Designer** scripting window.
4. Type the following where your cursor is currently positioned:
   ```javascript
   var itemID =
   ```
5. Select the **getDataValue** function from the **Functions** list. The **lawForm.getDataValue** string is added to the code automatically; it does not need to be entered manually.
6. Type “ITE-USER-FIELD1” inside the parenthesis. **Note:** The final string should look like this:
   ```javascript
   var itemID=lawForm.getDataValue("ITE-USER-FIELD1");
   ```
7. Press **Enter**.
8. Press **Tab**.
9. Type the following to enter the URL string:
   ```javascript
   var graingerURL = "http://www.grainger.com/Grainger/items/" + itemID + ""
   window.open(graingerURL)
   ```
10. Press **Enter**.
11. Press **Tab**.
12. Type **return true**;
13. Select the **Design** tab. **Note:** The **Design** tab is on the bottom of the **User Interface Designer** window.
14. Click Yes if prompted to Save Changes. Note: The script is not saved, but the changes are saved temporarily.
15. Select Save Selected, if prompted.

Part 3: Preview the form
2. Type 1234 in the Item Group field.
3. Type 1007 in the Item field.
4. Click Inquire. The IC11.1 form retrieves data.
5. Click the Grainger image logo. The Grainger website is launched and points directly to the item number defined on the string.
6. Click X to close the Grainer website.
7. Click X to close the preview window.

Part 4: Save the IC11.1 form
1. Select File > Save As.
2. Type IC11.1_<your initials>1.xml in the File name. Note: Add the number 1 after your initials to retain the original form as well as your new version.
3. Click OK.

Required fields
A common request using the Infor Lawson Design Studio is to make a field required. If the user leaves the required field blank during an action, such as Add or Change, a message displays and the focus is set back on the field.

Demo: Designate a required field
Your instructor will demonstrate how to designate a required field.
Exercise 5.2: Designate a required field

In this exercise, you will designate the ISO code as a required field on the Currency Code form (CU01.1). When a user performs a Change or an Add function on the form and neglects to include an ISO code, a message will appear prompting the user to enter an ISO code.

Exercise 5.2 steps

Part 1: Select the data area, system code, and form Id

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select File > New.
3. Select UI Designer.
4. Click OK.
5. Select APPS10.
6. Click Next.
7. Select IF (General Ledger Setup).
8. Click Next.
10. Click Next.
11. Confirm your selections.
12. Select QuickPaint.
13. Click Finish. The User Interface Designer window opens with a QuickPaint form.

Part 2: Capture the ISO Code textbox values

1. Click the ISO Code textbox field.
2. Click Data Src (...).
3. Write down the values for the ISO Code textbox field:
   - Data Src ____________________________
   - Name ______________________________
   - Field Nbr ____________________________

Part 3: Create the script for the OnBeforeTransaction function

Note: This function should be executed prior to sending the transaction.

1. Click the Script tab.
2. Select Form from the Objects menu.
3. Select OnBeforeTransaction from the Events menu. Note: This adds the function statement on the screen automatically.
4. Type the variable and statement **below** with the { } that were inserted from the step above:

```javascript
var continue_add=true
if(fc=="A"||fc=="C")
    continue_add=addCode()
if(continue_add)
    return true;
else
    return false;
```

**Part 4: Add a second function; addCODE() that triggers an alert message**

1. Type the following **below** the `OnBeforeTransaction` function.

```javascript
function addCode()
{
    var code=lawForm.getFormValue("text4");
    var message="You must enter an ISO CODE"
    if(code=="")
        return true;
    else
    {
        lawForm.setMessage(message);
        alert(message);
        return false;
    }
    return true;
}
```
Add the pre-built positionInField method to the alert (message). This sets the focus back on the ISO_Code field; the Field Nbr is noted in part 2, step 2 of this exercise.

```javascript
lawForm.positionInField("_f6");
```

The final addCode function should look like this:

```javascript
function addCode()
{
    var code=lawForm.getFormValue("text4");
    var message="You must enter an ISO CODE"
    if(code!=""
        return true;
    else
    {
        lawForm.setMessage(message);
        alert(message);
        lawForm.positionInField("_f6");
        return false;
    }
    return true;
}
```

3. Click the Design tab.
4. Click Yes when prompted to save.

**Part 5: Preview and save the file**

1. Select View > Preview. The CU01.1 preview form opens.
2. Select Canadian (CAD) in the Currency Code field.
3. Click Inquire.
4. Select the ISO Code field.
5. Clear the CAD entry.
6. Click Change. An alert message displays.
7. Click X to close the preview window.
8. Select File > Save As.
9. Type cu01.1_ex52_<your initials> in the textbox.
10. Click OK. The custom form is saved.
11. Type cu01.1.ex52_<your initials>.xml in the textbox.
12. Click OK. The XML file is saved.
13. Select File > Close to close the form.
Limiting a list box

JavaScript can be used to limit the options available in a user’s selection list.

Demo: Limit a listbox dynamically

Your instructor will demonstrate how to limit listbox dynamically.

Exercise 5.3: Limit a listbox dynamically

In this exercise, you will use JavaScript to limit clerks to pre-selected states in their listbox.

Exercise 5.3 steps

Part 1: Select the data area, system code, and form Id

1. Double-click Infor Lawson Design Studio from the desktop.
2. Select File > New.
3. Select UI Designer.
4. Click OK.
5. Select APPS10.
6. Click Next.
7. Select AP (Accounts Payable).
8. Click Next.
10. Click Next.
11. Confirm selection and select QuickPaint.
12. Click Finish. The User Interface Designer window opens with a QuickPaint form.

Part 2: Add a listbox and attach it to a data source

1. Click the Tab region on the AP10.1 form. Note: The Tab region is directly below the Attributes button. The Properties pane will display as tabregion1.
2. Select the Object tab (bottom of window). Note: Open the Object view when working with a Tab region.
3. Click the State or Province textbox and press Delete.
4. Double-click ListBox in the UI Toolbox.
5. Select the ListBox object.
6. Drag the ListBox object adjacent to the State or Province label.
7. Write down the following property values for the listbox:
   - Name___________
   - Field Nbr_________
8. Click Data Src (...).
9. Select VDR-STATE-PROV.
10. Write down the following property value for the listbox:
    - Field Nbr_________
    - Note: The Field Nbr value changed after selecting the data source. You will reference this field number to populate the listbox.
11. Click OK. The data source is assigned to the ListBox object.

Part 3: Add script to populate the listbox
1. Select the Script tab.
2. Select Form from the Objects menu.
3. Select OnInit from the Events menu. Note: This adds the function statement on the screen automatically.
4. Type the following script between the left and right curly brackets:
   
   ```javascript
   function FORM_OnInit()
   {
   lawformAddListboxOption('_f50','MN','Minnesota')
   }
   ```

5. Click the Design tab.
6. Click Yes when prompted to save.

Part 4: Preview and Save
2. Select Minnesota in the State or Province field.
3. Click X to close the preview window.
4. Select File > Save As.
5. Type ap10.1_ex53_<your initials> in the textbox.
6. Click OK.
7. Type ap10.1.ex53_<your initials>.xml in the textbox.
8. Click OK. Note: Do not close the file as you will add to it in the next exercise.
Demo: Populate a listbox by triggering a function from a button
Your instructor will demonstrate how to populate a listbox by triggering a function from a button.

Exercise 5.4: Populate a listbox by triggering a function from a button
In this exercise, you will use JavaScript to populate a listbox by triggering a function from a button. You will add to the ap10.1 form from the previous exercise.

Exercise 5.4 steps

Part 1: Add a button to the AP10.1 form
1. Double-click Button in the UI Toolbox.
2. Click Text (…).
3. Type Populate in the textbox.
4. Click OK.

Part 2: Add the script
1. Click the Script tab.
2. Type the following script below the OnInit function script's last curly bracket.

```javascript
function FORM_OnInit()
{
    lawformAddListboxOption('_f50','MN','Minnesota')
}

function PopulateList()
{
    lawformAddListboxOption('_f50','MN','Minnesota')
    lawformAddListboxOption('_f50','TX','Texas')
    lawformAddListboxOption('_f50','CA','California')
    lawformAddListboxOption('_f50','NJ','New Jersey')
    lawformAddListboxOption('_f50','IL','Illinois')

    alert('State values have been populated')
}
```

3. Click the Design tab.

4. Click Yes when prompted to save.

**Part 3: Assign function to button**

1. Select the Button object added in part 1, step 1 of this exercise.
2. Click Action (…).
3. Click the Function radio button.
4. Select the PopulateList function from the list.
5. Click OK.

**Part 4: Preview and save the file**

2. Click the Populate button. This triggers the PopulateList function.
3. Click OK. The message box closes.
4. Type 4321 in the Vendor Group field.
5. Click the Next button. Information for Vendor 1 displays on the screen.
6. Select IL-Illinois in the State or Province field.
7. Click the Change (pencil) button. The form returns a message ‘Change Complete – Continue’.
8. Click X to close the preview window.
9. Select File > Close to close the form.
10. Click Yes to save the changes. If you receive multiple save messages, click OK for each message box.
11. Click OK.

**Demo: Create an Extended Total button to Total Quantity Cost on detail lines (PO20.1)**

Your instructor will demonstrate how to create an Extended Total button to Total Quantity Cost on detail lines using the PO20.1 form.

**Exercise 5.5: Create an Extended Total button to Total Quantity Cost on detail lines**

In this exercise, you will use JavaScript to perform a calculation on the PO20.1 form after adding a new field to hold the total of the calculated values.

**Exercise 5.5 steps**

**Part 1: Select the data area, system code, and form Id**

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select File > New.
3. Select UI Designer.
4. Click OK.
5. Select APPS10.
6. Click Next.
7. Select Purchase Order (PO).
8. Click Next.
9. Select Purchase Order Entry (PO20.1).
10. Click Next.
11. Confirm the selections you made in steps, 5, 7, and 9.
12. Select QuickPaint.
13. Click Finish. The User Interface Designer window opens with a QuickPaint form.

Part 2: Add a textbox and a corresponding label

1. Double-click Textbox in the UI Toolbox.
2. Write down the Name of the textbox _______________ Hint: The Name is listed on the Properties pane, for example: text37.
3. Select the Textbox object.
4. Drag the Textbox object below the Header button but above the Item Detail section.
5. Double-click Label in the UI Toolbox.
6. Click Text (…).
7. Type Extended Total in the textbox.
8. Click OK.
9. Select the Extended Total Label object.
10. Drag the Extended Total Label object adjacent to the Textbox object in the User Interface Designer window.

Part 3: Create the script

1. Click the Script tab.
2. Select Form from the Objects menu.
3. Select OnAfterTransaction from the Events menu. Note: This adds the function statement on the screen automatically.
4. Type the variable and statement below:

```javascript
function FORM_OnAfterTransaction(val)
{
    var vTotal = 0
    for (var i=0;i<5;i++)
    {
        var vQty = lawForm.getDataValue("PLI-QUANTITY",i)
        var vCost = lawForm.getDataValue("PLI-VBUY-UNIT-CST",i)
        vTotal += vQty * vCost
    }
    vTotal = vTotal.toFixed(2)
    lawForm.getFormElement("text37").value = vTotal;
    return true;
}
```
5. Click the Design tab.
6. Click Yes to save the changes.
Part 4: Preview and save the form

2. Type 4321 in the Company field.
3. Click Next to see a numeric value in the Extended Total value field with each record.
4. Click X to close the preview window.
5. Select File > Save As.
6. Type po20.1_ex55 in the textbox.
7. Click OK. The custom form is saved.
8. Type po20.1.ex55_<your initials>.xml.
9. Click OK. The XML file is saved.
10. Select File > Close to close the form.
Working with multiple forms

JavaScript functions can be utilized to avoid adding duplicate information to multiple forms. For example, you could create a function that will update the PO Vendor (PO10.1) form automatically when a user executes an Add or Change on the Accounts Payable (AP10.0) form.

Demo: Update two forms simultaneously
Your instructor will demonstrate how to update two forms simultaneously.

Exercise 5.6: Update two forms simultaneously
In this exercise, you will update two forms simultaneously. You will use Company=4321 and Vendor=1000 for this exercise.

Exercise 5.6 steps

Part 1: Select the data area, system code, and form Id

1. Log in to Design Studio following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select File > New.
3. Select UI Designer.
4. Click OK.
5. Select APPS10.
6. Click Next.
7. Select Accounts Payable (AP).
8. Click Next.
9. Select Vendor (AP10.1).
10. Click Next.
11. Confirm your selection and select QuickPaint.
12. Click Finish. The User Interface Designer window opens with a QuickPaint form.
Part 2: Create the script

1. Click the **Script** tab.
2. Select **Form** from the **Objects** menu.
3. Select **OnAfterTransaction** from the **Events** menu. This function is triggered once a user executes the Add or Change action on AP10.1.
4. Type the following variable and statement between the left and right curly brackets of the **OnAfterTransaction** function:

   ```javascript
   function FORM_OnAfterTransaction(data)
   {
     if(formState.agsError) return;

     if(data.selectSingleNode("//_f1").text=="A"||
        data.selectSingleNode("//_f1").text=="C")
     {
       var sVendorGroup = lawForm.getDataValue("VEN-VENDOR-GROUP");
       var sVendorEmail = lawForm.getDataValue("VEN-MAIL-ADDRESS");

       if(data.selectSingleNode("//_f1").text=="A")
       {
         var sVendor = lawForm.getMessage();
         sVendor = sVendor.substr(sVendor.indexOf(" ")+1, sVendor.length);
         sVendor = sVendor.substr(0, sVendor.indexOf(" "));
       }
       else
       sVendor = lawForm.getDataValue("VEN-VENDOR")

       var sAGSCall = strAGSPath + "?_PDL=";
       sAGSCall += portalWnd.oUserProfile.getAttribute("productline");
       sAGSCall += "&_TKN=PO10.1&EVT=ADD&_LFN=ALL&_TDS=IGNORE&FC=A";
       sAGSCall += "&POV-VENDOR-GROUP=" + sVendorGroup;
       sAGSCall += "&POV-VENDOR=" + sVendor;
       sAGSCall += "&POV-EMAIL-ADDRESS=" + sVendorEmail;
       sAGSCall += "&_OUT=XML&_EOT=TRUE";

       var sAGSInfo = top.httpRequest(sAGSCall);
       if(!sAGSInfo||sAGSInfo.status>400)
   ```
\{ 
    alert("Page Not Found")
    return;
\} 

var sFldNbr = sAGSInfo.selectSingleNode("//FldNbr").text;
var sMsgNbr = sAGSInfo.selectSingleNode("//MsgNbr").text;
var sMessage = sAGSInfo.selectSingleNode("//Message").text;
if(sMsgNbr=="000")
    alert("Good Job!")
else
    {
        sAGSCall = sAGSCall.replace("EVT=ADD","EVT=CHG")
        sAGSCall = sAGSCall.replace("FC=A","FC=C")
        var sAGSInfo = top.httpRequest(sAGSCall);
        var sFldNbr = sAGSInfo.selectSingleNode("//FldNbr").text;
        var sMsgNbr = sAGSInfo.selectSingleNode("//MsgNbr").text;
        var sMessage = sAGSInfo.selectSingleNode("//Message").text;
        if(sMsgNbr=="000")
            alert("Good Job!")
        else
            alert(sFldNbr + ", " + sMessage + ", " + sMsgNbr)
    }
    alert(sFldNbr + ", " + sMessage + ", " + sMsgNbr)
  
  
5. Click the Design tab.
6. Click Yes if prompted to save.

Part 3: Preview and test the form
2. Type 4321 in the Vendor Group field.
3. Type 1000 in the Vendor field.
4. Click Inquire.
5. Click the Contact tab.
6. Type <your email address> in the E-mail field.
7. Click Change. The “Good Job” message displays.
8. Click OK to close the message box and return to the home page.

Lawson: Customizing Lawson Applications with Design Studio and JavaScript Training Workbook 93
9. Type \textit{PO10.1} in the search box (upper right hand corner).

10. Press \textbf{Enter}.

11. Click \textbf{Inquire}. The email field is updated with your email address.

12. Click \textbf{X} to close the preview windows.

\textbf{Part 4: Save the form}

1. Select \textbf{File > Save As}.

2. Type \textit{AP10.1\_ex56\_<your initials>} in the textbox.

3. Click \textbf{OK}. The custom form is saved.

4. Type \textit{ap10.1.ex56\_<your initials>.xml} in the textbox.

5. Click \textbf{OK}. The \textbf{XML file} is saved.

6. Select \textbf{File > Close} to close the form.

\begin{itemize}
\item \textbf{Demo: Add HR data to an Accounts Payable (AP) form}
\item Your instructor will demonstrate how to add HR data to an Accounts Payable (AP) form.
\end{itemize}

\begin{itemize}
\item \textbf{Exercise 5.7: Add HR data to an Accounts Payable (AP) form}
\item In this exercise, you will add HR data to an Accounts Payable (AP) form. For this exercise, we will assume that the employee is also a vendor.
\end{itemize}

\textbf{Exercise 5.7 steps}

\textbf{Part 1: Select the data area, system code, and form Id}

1. Log in to \textbf{Design Studio} following the steps 1-4 in exercise 3.1 if you are not logged on to the Design Studio application.

2. Select \textbf{File > New}.

3. Select \textbf{UI Designer}.

4. Click \textbf{OK}.

5. Select \textbf{APP10}.

6. Click \textbf{Next}.

7. Select \textbf{Accounts Payable (AP)}. 
8. Click **Next**.
9. Select **Vendor (AP10.1)**.
10. Click **Next**.
11. Confirm your selection and select **QuickPaint**.
12. Click **Finish**. The **User Interface Designer** window opens with a QuickPaint form.

**Part 2: Delete textbox fields and corresponding buttons to allow room for three labels and three textbox fields**

1. Click **Attributes**.
2. Press **Delete**.
3. Click the **Textbox** object adjacent to the button. **Hint**: The **Textbox** object will display a name, such as _out4_, in the **Properties** pane.
4. Press **Delete**.
5. Repeat steps 1-4 with the following buttons and textboxes: **Hint**: Press and hold the Ctrl key to select multiple items.
   - Vendor Bank
   - Customer
   - Approval
   - Locations

**Part 3: Add three textboxes with corresponding labels to the form**

1. Double-click **Textbox** in the **UI Toolbox**.
2. Write down the **Field Nbr** ____________________ and **Name** ____________________.
3. Double-click **Label** in the **UI Toolbox**.
4. Select the **Label** object.
5. Drag the **Label** object adjacent to the **Textbox** object.
6. Arrange the textbox and label on the form above the **Main** section. **Hint**: This is the same place where the buttons were that you had deleted in part 2 of this exercise.
7. Select the **Label** object.
8. Click **Text (...)**.
9. Type **Employee Number** to label the object in the textbox.
10. Click **OK**.
11. Repeat steps 1-8 to add two more textboxes and corresponding labels: **Department**, **Position**.
    Remember to note the **Field Nbrs** and **Names** for these as well.

    - Department Field Nbr ____________________ Name ____________________
    - Position Field Nbr ____________________ Name ____________________

**Part 4: Create the script**

1. Click the **Script** tab.
2. Select **Form** from the **Objects** menu.
3. Select **OnAfterTransaction** from the Events menu.
4. Type the following variable and statement between the left and right curly brackets of the `OnAfterTransaction` function:

```javascript
function FORM_OnAfterTransaction(data)
{
    // was transaction successful?
    if(formState.agsError) return;
    var vVen = lawForm.getDataValue("VEN-VENDOR");
    var vCompany = lawForm.getDataValue("VEN-VENDOR-GROUP");
    //alert(vVen);

    // Do DME call to get employee information from HR11.1 (Employee Table)
    var s = portalWnd.DMEPath;
    s += "?PROD=" + strPDL;
    s += "&FILE=EMPLOYEE&INDEX=EMPSET1&KEY=" + vCompany + "+" + vVen;
    s += "&FIELD=EMPLOYEE;DEPARTMENT.NAME;POSITION.DESCRIPTION";
    s += "&OUT=XML&MAX=1";

    // Your field numbers may not match the numbers below.
    lawForm.setFormValue("text74","");
    lawForm.setFormValue("text75","");
    lawForm.setFormValue("text76","");

    // send the DME call to the server
    var vDMEInfo = portalWnd.httpRequest(s);

    if(vDMEInfo || vDMEInfo.status)
    {
        var msg = "Error calling DME,";
        msg += (vDMEInfo
            ? "(status code): " + vDMEInfo.status
            : "bad server response.");
        alert(msg);
        return true;
    }
}
```
//create an XML object to contain the DME data
var vObjDMEXML = new portalWnd.DataStorage(vDMEInfo);

//load a variable with the records returned by DME
var vRecords = vObjDMEXML.document.getElementsByTagName("RECORD");
if(vRecords.length == 0)
    return true;

//Display information on the page. Note, ids may change on your custom form.
var vCols = vRecords[0].getElementsByTagName("COL");
lawForm.setElementValue("_f184",vCols[0].firstChild.data);
lawForm.setElementValue("_f185",vCols[1].firstChild.data);
lawForm.setElementValue("_f186",vCols[2].firstChild.data);

5. Click the Design tab.
6. Click Yes if prompted to save.

Part 5: Preview and test the form
2. Type 4321 in the Vendor Group field.
3. Type 1000 in the Vendor field.
4. Click Inquire. The Employee Number, Department and Position is populated on the form.
5. Click X to close the preview window and return to the User Interface Designer window.

Part 6: Save the form
1. Select File > Save As.
2. Type AP10.1_ex57__<your initials> in the textbox.
3. Click OK.
4. Type AP10.1_ex57__<your initials>.xml in the textbox field.
5. Click OK.
6. Type X to close the form. The XML file is saved.
Enhancing a custom page with JavaScript

Customers who have JavaScript expertise can add enhancements such as interactivity to custom pages. By accessing **Object Viewer > Pages** section, available via the Tools menu, you can view reference materials on Custom Page Designer objects that can be accessed through the scripting interface.

You can create Transaction, Data and Drill strings using **API Builder** that can be used to access the Infor Lawson data from within the Scripting environment or can be pasted into other applications for use.

---

**Demo: Use Custom Page Designer to create a custom page to approve invoices**

Your instructor will demonstrate how to use Custom Page Designer to create a custom page to approve invoices.

---

**Exercise 5.8: Use Custom Page Designer to create a custom page to approve invoices**

In this exercise, you will learn how to use Custom Page Designer to create a custom page to approve invoices.

**Exercise 5.8 steps**

**Part 1: Add a data query object to a custom page**

1. Log in to **Design Studio** following the steps 1-4 in exercise 3.1, if you are not logged on to the Design Studio application.
2. Select **Custom Page Designer**.
3. Click **OK**. The **Custom Page Designer** window opens.
4. Click the **Design** tab if not already open.
5. Double-click **Data Query** from the **Lawson Toolbox**.
6. Type or select the property values specified in this table for the **Data Query** object:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>ap_invoice_approval_data0</td>
</tr>
</tbody>
</table>

---

98 Lesson 5: Enhancing Infor Lawson Design Studio with JavaScript
7. Click **Custom (…)**.

**Part 2: Build the query**

1. Click **AP (…)**. The API Builder is launched.
2. Type or select the data specified in the table in the API Builder:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Queries</td>
<td>APPS10</td>
</tr>
<tr>
<td>System Code</td>
<td>Accounts Payable (AP)</td>
</tr>
<tr>
<td>Data Tables</td>
<td>APINVOICE</td>
</tr>
<tr>
<td>Table Elements</td>
<td>COMPANY, VENDOR, INVOICE, VOUCHER-NBR, AUTH-CODE</td>
</tr>
<tr>
<td>Indices</td>
<td>APISET1</td>
</tr>
<tr>
<td>Index Elements</td>
<td>Company</td>
</tr>
<tr>
<td></td>
<td>4522</td>
</tr>
</tbody>
</table>

3. Click **Add Criterion**.
4. Select **INV-STATUS**.
5. Select **= equals**.
6. Select **Literal**.
7. Type **Unreleased** in the textbox.
8. Select the **Ignore Case** checkbox.
9. Click **Add**.
10. Click **Build Query**. The data query is created.
11. Click **OK** to return to the **Data Query Property Pages**.

**Part 3: Add a button**
1. Click the **Fields** tab.
2. Click the **Body** tab.
3. Select the **Button** radio button.
4. Type **Release** in the **Button Title** field.
5. Type **Release** in the **Button Value** field.
6. Click **Apply**.
7. Click **OK**.

**Part 4: Preview the custom page**
1. Select **View > Preview**. The data displays for **company 4522** and the **Release** button is enabled.
2. Click **X** to close the preview window and return to the **Custom Page Designer** window.

**Part 5: Map the data query object**
1. Click the whitespace in the **Custom Page Designer** window. **Hint:** Place the cursor above the **ap_invoice_approval_data0** object and check that the **Properties** pane **Name** displays as **portalpage**.
2. Click **Custom (...)**. The **Data Map** window opens.
3. Select the **ap_invoice_approval_data0** from the **Objects** drop-down list.
4. Select **Company** from the left panel.
5. Click the **right-facing arrow** to move it to the right panel.
6. Repeat step 5 for the following:
   - AUTH-CODE
   - INVOICE
   - VENDOR
   - VOUCHER-NBR
7. Click **Apply**.
8. Click **OK** to return to the **Custom Page Designer** window.

**Part 6: Add a text object to the custom page**
1. Click **General** from the **Toolbox** menu.
2. Double-click **Text**.
3. Type or select the property values specified in this table for the text object:
4. Click **Custom (…)**. This opens the **Text Property Page**.

5. Type **Invoice Approval** in the text field.

6. Click **Apply**.

7. Click **OK**.

### Part 7: Add script

1. Click the **Script** tab.
2. Select **ap_invoice_approval_data0** from the **Objects** menu.
3. Select **OnClickButton** from the **Events** menu. This adds the **OnClickButton** script to the page.
4. Type the following variable and statement between the left and right curly brackets of the **OnClickButton** function.

```javascript
function ap_invoice_approval_data0_OnClickButton(dmId, row)
{
    //create variables for data mapped from page
    var vServer = window.location.host
    var vCompany = page.dataSource["COMPANY"];
    var vAuthCode = page.dataSource["AUTH-CODE"];
    var vVoucher = page.dataSource["VOUCHER-NBR"];
    var vVendor = page.dataSource["VENDOR"];
    var vInvoice = page.dataSource["INVOICE"];
    var vProd = portalWnd.lawsonPortal.getUserVariable("productline");
    //simple check to see if a row was clicked and data exists
    if(vCompany=="")
    {
        alert("Please select an invoice");
    }
}
```

© 2018 Infor Education. All rights reserved. No part of this document may be reproduced or transmitted in any form, without written permission.
return;
}  
// space fill vendor field
var nLength = vVendor.length;
if(nLength<9)
{
    for(n=1;n<(10-nLength);n++)
        vVendor = "%20" + vVendor;
}

nLength = vVoucher.length;
if(nLength<10)
{
    for(n=1;n<(11-nLength);n++)
        vVoucher = "%20" + vVoucher;
}

// example of the original ap26 api
var s = portalWnd.AGSPath + "?_PDL=" + vProd + 
"&_TKN=AP26.1&_LFN=ALL&_EVT=CHG&_RTN=DATA&_TDS=IGNORE&_OUT=XML&_FC=C" 
+ "&_API-COMPANY=" + vCompany 
+ "&_PT-API-VOUCHER-NBR=" + vVoucher 
+ "&_PT-API-AUTH-CODE=" + vAuthCode 
+ "&_LINE-FCr0=R" 
+ "&_API-AUTH-CODEr0=" + vAuthCode 
+ "&_API-VENDORr0=" + vVendor 
+ "&_API-INVOICER0=" + vlnvoice 
+ "&_EOT=TRUE";

// send the api to the server
var oResponse=top.httpRequest(s);
if(!oResponse||oResponse.status)
    return;

// extract out a specified node from the return object
try
{
    var vMessage = oResponse.selectSingleNode("//Message").text;
}
catch(e){}
alert(vMessage + " - Now Refreshing");

//refresh the page
document.location.reload();

}

5. Click the Design tab.
6. Select Yes if prompted to save.

Part 8: Preview and save the custom page

1. Select View > Preview. The data displays for company 4522.
2. Click Release for the following 4522 company invoices:
   - INVOICE1007A
   - INVOICE1007B
   - INVOICE1007C
   - INVOICE1007D
3. Click X to close the preview window.
4. Select File > Save As. Note: If prompted, select Remote instead of Local.
5. Type ex58__<your initials> in the textbox field.
6. Click OK. The custom page is saved.
7. Click OK to save the ap_invoiceApproval_data0.xml file.
8. Click OK to save the ap_invoiceApproval_text0.xml file.
Check your understanding

A set of multiple values for a field can be included in a form using the toolbox object called:

a) Checkbox   
b) Listbox   
c) Tab area

The ______ is a tool that provides reference material about Infor Lawson objects that can be accessed through the scripting interface.

a) API Builder   
b) Custom button   
c) Object Viewer

________ programming extends the functionality of the Infor Lawson Design Studio.

a) Custom Page Designer   
b) Wizard Designer   
c) JavaScript

A ______ is an action that you can associate with an object.

a) Event   
b) Data Object   
c) Function

The ______ view shows the transaction fields.

a) Design   
b) Data Object   
c) Function

________ provide a way for developers to group functions so they can be re-used within customizations.

a) Include Files   
b) Objects Viewer   
c) Events
Which three of the following sections are part of the Object Viewer?

a) List view  
b) Infor Lawson for Infor Ming.le  
c) Form  
d) Pages

Which function would you use to create a required field?

a) TABREGION  
b) OnBeforeTransaction  
c) reqCODE

Which function would you use to combine multiple functions, such as adding a record to two applications?

a) OnAfterTransaction  
b) OnBeforeDataExchange  
c) getNodeValue

Which object would you use in Custom Page Designer to add a button on a custom page for approving invoices?

a) RSS news  
b) Report  
c) Data query
Course summary

Estimated time
1 hour

Learning objectives
Now that you have completed this course, you should be able to:

- Define the process for creating help text and "stepping" users through forms using Wizard Designer.
- Identify the process for customizing Infor Lawson forms using User Interface (UI) Designer.
- Define ways to allow automation of form actions such as add, delete, and inquire.
- Describe the process for using the Custom Page Designer to create custom pages.
- Explain how to use Infor Lawson API Builder to build the string used to retrieve and send Infor Lawson data through the Internet Object Services (IOS) engine.
- Define how to extend the functionality of a form using JavaScript™.
- Describe how to use Infor Lawson Design Studio’s JavaScript development environment to add customized functionality to a form or custom page.

Topics
- Course review
Course review

_________ is an Infor Lawson Design Studio component that lets users customize Infor Lawson application forms.
   a) Custom Page Designer
   b) Wizard Designer
   c) UI Designer

_________ creates a string to retrieve and send Infor Lawson data through the Internet Object Services engines.
   a) JavaScript
   b) API Builder
   c) Portal

A _______ is a utility attached to an Infor Lawson form that provides step-by-step instructions on how to complete a specific task.
   a) Wizard Designer
   b) Menu
   c) API

The Wizard Designer utilizes _________ and associates them with fields to build wizards.
   a) Scripts
   b) Steps
   c) Menus

Which of the following are examples of form customizations using UI Designer? Select all that apply.
   a) Changing the tabbing order on a form
   b) Removing tabs from a form with multiple tabs
   c) Removing fields
   d) Adding images or logos to a form
   e) Hiding fields
________ are aspects or values of the object that you can change such as text, font color, width size, and default value.

a) Properties  
b) Toolbox  
c) Objects

Which of the following are linked objects? Select all that apply.

d) Web page  
e) Image  
f) Formlet  
g) Report

Which of the following are examples of Special objects in Custom Page Designer? Select all that apply.

a) Formlet  
b) Text  
c) Composite  
d) Mapping

Which object would you use in Custom Page Designer to add a button on a custom page for approving invoices?

a) RSS news  
b) Report  
c) Data query

________ allows designers to create their own customized tags, enabling the definition, transmission, validation and interpretation of data between applications and between organizations.

a) Forms  
b) Objects  
c) XML
True or False: UI Designer (without custom script) can change the behavior of an Infor Lawson form.

   a) True
   b) False

The______ provides information about Infor Lawson objects and methods for JavaScript developers.

   a) Wizard Designer
   b) Object Viewer
   c) JavaScript