PeopleSoft

PeopleSoft 8.9
J.D. Edwards Web Client
PeopleBook

September 2003

Table of Contents

Web Application User Interface

Most J.D. Edwards applications are available on the Web. These applications provide the same functions as their Windows counterparts, although the interface differs somewhat due to differences between the two platforms.

J.D. Edwards Web-based applications are powered by the JAS server, which a system administrator configures when implementing the J.D. Edwards system. However, Web and Windows applications both reference the same database tables for the most part, so Web and Windows users can use the same system simultaneously, and both can see all changes in real time, depending on the constraints of your system. Additionally, the JAS server can communicate with pervasive devices, so you can access certain J.D. Edwards applications from such devices as well.

A user can log on to the Web client directly, or she can log on through the J.D. Edwards Portal, depending on how a system administrator configures J.D. Edwards ERP security and whether the Portal is implemented. In either case, users access J.D. Edwards applications through Task Explorer. Users who log in to the Web client directly see Task Explorer as a stand-alone interface, while users who log on through the Portal access Task Explorer through a Portal component. Task Explorer is the Web counterpart of the Windows-based Solution Explorer, and it provides many of the same features.

Accessing the J.D. Edwards Web Client

Accessing J.D. Edwards software on the Web is similar to accessing files and applications on a company network. Typically, either when you start your computer or when you want to access company network directories, you must log on to identify yourself to the system as an employee who has the right to access company resources. After you log on, you can access files and applications on the network. Similarly, you must log off the Web client—the gateway through which you access J.D. Edwards software—before you can launch J.D. Edwards applications.

To log on to the Web client, your computer must have access to your company's intranet, you must have a Web browser installed on your computer, and your system administrator must create an account for you. If you are using a pervasive device, however, all you need is an account. When your system administrator sets up your account, he or she creates a user ID and assigns you a password.

Usually, you must enter your ID and password when you launch the Web client. However, your system administrator can configure your computer in such a way so that you appear to bypass the logon process. For security reasons, however, most system administrators want users to log on manually.

When you log on to the Web client, Task Explorer appears. Task Explorer allows you to access J.D. Edwards applications, reports, and other features.

Although logging on to the J.D. Edwards Web client is similar to logging on to your company's network, logging off is different. In most cases, to log off a network connection you simply turn off your computer. However, after you have logged on to the J.D. Edwards Web client, you should actually log off when you are finished. That is, just closing the Web browser or shutting off your computer or pervasive device is insufficient. Before you close your Web browser and shut off your computer, you must complete the logoff procedure.

Note

You might access Task Explorer through J.D. Edwards Portal. If you do, then you do not need to log on to Task Explorer because you have already logged on to Portal. Similarly, you do not need to log off Task Explorer, although you should log off Portal when you are finished with it.

► To log off the J.D. Edwards Web client

On Task Explorer, click Logout.

► To log on to the J.D. Edwards Web client

1. Launch your Web browser and navigate to your company's J.D. Edwards Web login.

   Depending on how your system administrator has configured your system, the Web login might appear when you launch your browser, you might need to click a button or a hyperlink, or you might need to navigate to a particular page. If you do not know how to find the J.D. Edwards Web login, contact your system administrator.

2. Complete the following fields:
   - User ID
   - Password
3. If your system administrator indicated that you must log in to a particular environment, click Login Details and complete the following field:
   - Environment
4. Click Login.

   The Task Explorer appears.

Task Explorer

Task Explorer runs in your Web browser and allows you to access J.D. Edwards applications, reports, and other features. Task Explorer appears automatically after you log in to the J.D. Edwards Web client, either on a workstation or a supported pervasive device. You can also access Task Explorer from J.D. Edwards Portal through the Task Explorer component.

If you access the Task Explorer directly (in other words, you do not access the Task Explorer through the J.D. Edwards Portal), a button that says Logout appears in the upper right. Click Logout when you are finished with Task Explorer.

At the top of Task Explorer is a bar which might contain one or more hyperlinks, buttons, or drop-down menus. The main portion of Task Explorer displays a tree that you can use to navigate to the specific application that you want to launch. The tree can contain other objects in addition to applications, such as shortcuts and links. For this reason, all objects in the tree are called tasks. In other words, folders, applications (including reports), links, and shortcuts in the tree are all tasks. Each time you click a folder, you expand the tree a level and the view of the tree changes.

Principal sets of tasks are called task views. When you log in, Task Explorer displays your default task view if you have one defined. Otherwise, the system displays the OneWorld Menus task view. You cannot navigate from one task view to another unless a task view includes a shortcut to another one. Typically, you use the Task View field to switch between task views, but depending on how your system administrator has configured your system, you might not be able to see the Task View field.

Do not confuse task views with the new view of a tree that occurs when you expand it. Although your view of the tree changes, you have not actually entered a new task view. Task Explorer documentation refers to the changing tree views as views.

System administrators use Solution Explorer and J.D. Edwards ERP security applications to manage user accounts and to configure Task Explorer.

See Also

- *Initial Configuration* and related topics in the *Solution Explorer Guide* for more information about configuring task views
- *Solution Explorer Security* in the *System Administration Guide* for more information about configuring Task Explorer functions that are available to end users

► To change your default task view

1. In Task Explorer, choose the task view you want to use as your default task view from the drop-down menu in the Task View field.
2. Click Save As Default.

► To navigate in Task Explorer

1. In Task Explorer, choose a task view from the drop-down menu in the Task View field.

   The tree displays the root tree structure for the task view.

2. Click a task marked with a folder icon.

   The tree displays only the tasks under the task you clicked. Task Explorer appends the name of the current view to the name of the previous view and makes the previous name a hyperlink. As you navigate deeper into the task view structure, Task Explorer continues to append the name of the current view onto the title, while turning the previous view names into hyperlinks. This display provides a record of how you reached the current view and allows you to return to any view in the progression.

3. To return to a previous view in your current progression, click its hyperlink.

   Note
   Do not use the Back button on your browser to return to a previous task view.

Fast Path

Fast Path is a field that allows you to access a specific task (that is, folder, application, or report) directly. You use commands in Fast Path to quickly move among menus and applications. A fast-path command can be any of the following items:

- An abbreviation that is either shipped with J.D. Edwards demo data or that you define to suit your business environment. For example, the code BV might access the Batch Versions application so that you can run a report.
- A task ID.
- A program name.

To use the Fast Path field, enter a Fast Path code and click the button to the right of the field.

Depending on how your system administrator configured your account, you might not be able to see or change your Fast Path security.

In Task Explorer, you can use Fast Path codes to launch task views, J.D. Edwards applications, and so on. To specify a task view, enter TV: followed by its internal task ID. For example, TV:98 accesses your Favorites task view.

You can also use the Fast Path field to access menus. Task views are composed of menus and individual tasks. Menus have no special format in Task Explorer; they simply provide application developers with a convenient method of grouping applications. When you access a menu, you actually access a specific place in a task view. To access a menu, enter its ID. For example, G0 accesses the Foundation Systems menu.

To launch an application, enter the application's program number. To specify a form in the application, enter the application's program number followed by a |, and then enter the form ID. For example, when you enter P01012|W01012B, the system displays the Work with Addresses form in the Address Book application. You can specify a version of a form to open by adding the version number after the form name with a |, for example, P01012|W01012B|ZJDE0003.

Contact your system administrator for specific internal task, menu, and application IDs.

Note

Not all objects have Fast Path commands.

► To launch an application or report

1. In Task Explorer, navigate to the application or report you want to launch.
2. To launch the application or report without defining processing options, version, and so forth, click the report or application.

   Applications launch immediately. If you launch a report without choosing a version, the system launches Work with Batch Versions so you can choose which version you want to run.

3. To select processing options or version for an application, hover your cursor over the green arrow next to the task and choose Values or Versions, respectively.

   If a menu option is italicized, then that option is unavailable for that application.

   After you choose the processing options or version, the system launches the application.

4. To select processing options or version or to designate data selection parameters for a report, hover your cursor over the green arrow next to the task and choose one of the following options:

    - Values
    - Versions
    - Data Selection
    - Data Selection & Values

    If a menu option is italicized, then the option is unavailable for the report.

    After you set the options, the system might launch Work with Batch Versions so you can choose which version you want to run. Then, Version Prompting appears. Choose the prompting you want and click Submit to choose a printer and process the report.

See Also

- *Interactive Versions for Applications* in the *Foundation Guide* for more information about running J.D. Edwards ERP applications
- *Batch Versions for Reports* in the *Foundation Guide* for more information about running J.D. Edwards reports
- *Processing Options* in the *Foundation Guide* for more information about processing options and how to use them

Favorites Task View

The Favorites task view is a task view where you can save links to other tasks. If you frequently run a task or navigate to a particular place, you can save that task or place in your favorites list. Then, you can access that task directly from your Favorites task view.

You have your own Favorites task view, and the other users in your company have their own Favorites task views. No one else can see your Favorites task view or your changes. If you also use Solution Explorer, the Windows version of Task Explorer, then you see the same favorites list in both applications, provided that you log into the same environment.

Depending on how your system administrator configured your account, you might not be able to see or change your Favorites task view.

► To access your Favorites task view

In Task Explorer, choose Favorites from the drop-down list in the Task Views field.

► To add a task to your Favorites task view

1. In Task Explorer, navigate to the task that ou want to add to your Favorites task view.
2. Hover your cursor over the arrow next to the task and choose Add To Favorites.

Role-Based Task Views

Your system administrator can create task views that are available to users with specific roles. Roles are a way to categorize and group users. You system administrator assigns you one or more roles when creating your account. Role-based task views can provide a convenient and succinct list of applications and reports necessary to perform a certain job.

► To access a role-based task view

1. In Task Explorer, click Task Views and choose a role-based task view such as End-User Tasks.

   The task views that appear are those which are valid for your default role.

2. To change your role and thereby see a different selection of role-based task views, click Roles and choose a role.

   The roles that appear are those to which you have been granted access by your system administrator.

User Options

When you click User Options in Task Explorer, the User Default Revisions form appears. The following list describes the associated action for each button on the User Default Revisions form:

Button	Description
User Profile Revisions	Launches the User Profile Revisions program (P0092). Only system administrators should change user profiles.
View Local Output	Accesses the PrintQueue directory on the machine that is running J.D. Edwards software.
Submitted Reports	Launches the Work With Servers program (P989116), which you can use to review the status of a submitted report or job, change your report or job priority, work with the report output, and review errors.
Menu Revisions	Has no effect in Task Explorer.
Change Password	Launches the OneWorld Security program (P98OWSEC), which you use to change your password.
Default Printer	Launches the Printer Application program (P98616). Only system administrators should change default printer settings.

See Also

- *User Profiles* in the *System Administration Guide* for information about modifying user profiles
- *To View Report Output* in the *Foundation Guide* for information about accessing the PrintQueue directory
- *Submitting a Report* in the *Enterprise Report Writing Guide*
- *To change your password* in the *JD Edwards Web Client Guide*
- *Working with the Printers Application* in the *Enterprise Report Writing Guide* for information about changing default printer settings

► To change your password

1. In Task Explorer, click User Options.
2. On User Default Revisions, click Change Password.
3. On User Password Revisions, complete the following fields and click OK:
   - Old Password
   - New Password
   - New Password - Verify

► To view report output

Before you can view the output of your reports online, you must run a report version.

1. In Task Explorer, click User Options.
2. On User Default Revisions, click View Local Output.
3. On the Open form, choose a file and click Open.

   A PDF version of the report appears. You can also view log files, such as error logs. To do so, choose UBE Log Files from the Files of Type field.

See Also

- *Submitting a Report* in the *Enterprise Report Writing Guide* for information about how to run a report

Task Explorer Configuration

As a system administrator, you must perform the following tasks to set up and maintain Task Explorer:

- Set up Solution Explorer on a Windows workstation
- Configure Task Explorer to reference the correct Explorer tables
- Create roles with Solution Explorer
- Create task views with Solution Explorer
- Set up security with Security Workbench

Even if your Windows users are still using OneWorld Explorer to access J.D. Edwards applications, you can configure Task Explorer roles and task views only with Solution Explorer. If your Windows users are running OneWorld Explorer, then you should point Task

Explorer to the various OneWorld Menu tables (F008x). On the other hand, if your Windows users are running Solution Explorer, you should point Task Explorer to the various Solution Explorer tables (F900x). If you have implemented J.D. Edwards ERP software, OneWorld Xe, or ERP 8.0, J.D. Edwards recommends that you use Solution Explorer and its associated tables. If you use the OneWorld Explorer tables, then your users can access only one task view (OneWorld Menus).

To facilitate switching between these table sets, J.D. Edwards provides two template files: `taskexplorer.properties.owmenu` and `taskexplorer.properties.activera`. You overwrite the contents of `taskexplorer.properties` with the contents of `taskexplorer.properties.owmenu` to point to the OneWorld Menu tables. You overwrite the contents of `taskexplorer.properties` with the contents of `taskexplorer.properties.activera` to point to the Solution Explorer tables. Initially, `taskexplorer.properties` points to the Solution Explorer tables.

Role-based task views are one of the most powerful features of Solution and Task Explorer. To set up role-based task views, you must first set up roles. Then you can create or modify task views tailored to your users' needs. You use Solution Explorer to set up roles and task views. Refer to the Solution Explorer documentation for details about setting up roles and task views. Both of these features are dependent on the Solution Explorer tables, so if you are pointing to the OneWorld Menu tables, you cannot take advantage of roles or multiple task views.

Security Workbench includes a form for Solution Explorer (and therefore Task Explorer) security options, although only the options Explorer, Favorites, and Fast Path/Find It! affect Task Explorer. You can allow users to view secured task views. You can allow users to see or to modify their own Favorites list. You also can allow users to access the Fast Path feature (Find It! is currently unavailable in Task Explorer). You can apply different security configurations to roles or to individual users.

The system references UDC codes in UDC type Fast Path Commands (00/FP) for Fast Path if Task Explorer is pointing to the OneWorld tables. If Task Explorer is pointing to the Solution Explorer tables, the system references UDC codes in UDC type ActivEra FastPath (H90/FP) instead.

See Also

- *Solution Explorer Security* in the *System Administration Guide* for details about allowing users access to Task Explorer task views, the Favorites task view, and Fast Path
- *Setting Up User Roles* in the *System Administration Guide* for details about creating and administering roles
- *Task View Set Up* in the *Solution Explorer Guide* for details about creating and modifying task views
- *Task Set Up* in the *Solution Explorer Guide* for details about creating and modifying tasks to include in task views and about applying roles to a task

► **To configure Task Explorer to reference the correct Explorer tables**

1. Stop the J.D. Edwards instance of WebSphere.
2. On the Web server, go to \\..\webclient\taskexplorer\.

   The file `taskexplorer.properties` governs how Task Explorer runs, including which set of tables it uses. The default is for Task Explorer to point to the Solution Explorer tables.

   Caution
   Do not modify `taskexplorer.properties` except as instructed below.

3. Rename `taskexplorer.properties` to `taskexplorer.properties.def`.
4. To point Task Explorer to the OneWorld Menu tables, rename `taskexplorer.properties.owmenu` to `taskexplorer.properties`.
5. To point Task Explorer to the Solution Explorer tables, rename `taskexplorer.properties.activera` to `taskexplorer.properties`.
6. Start the J.D. Edwards instance of WebSphere.

Troubleshooting Task Explorer

If you are an end user and are having problems with Task Explorer, contact your IT support professional or your system administrator.

As a system administrator, if Task Explorer is not appearing, review the following logs:

- Examine the JAS.log. Ensure that all the drivers have been registered. Look for SQL (or equivalent) exceptions to track errors in database connectivity.
- Set DEBUG=TRUE in the JAS.ini file and examine the JASDEBUG.log for errors in API activity.
- Examine STDOUT.log and STDERR.log. STDOUT.log in particular contains Task Explorer-specific logging such as errors related to entry point form lookups.

Log location is determined by the JAS.ini file.

J.D. Edwards Web Applications and Reports

J.D. Edwards provides a variety of applications, reports, and other objects. Typically, you access these objects from the Task Explorer.

► **To launch an application or report**

1. In Task Explorer, navigate to the application or report you want to launch.

2. To launch the application or report without defining processing options, version, and so forth, click the report or application.

   Applications launch immediately. If you launch a report without choosing a version, the system launches Work with Batch Versions so you can choose which version you want to run.

3. To select processing options or version for an application, hover your cursor over the green arrow next to the task and choose Values or Versions, respectively.

   If a menu option is italicized, then that option is unavailable for that application.

   After you choose the processing options or version, the system launches the application.

4. To select processing options or version or to designate data selection parameters for a report, hover your cursor over the green arrow next to the task and choose one of the following options:

   - Values
   - Versions
   - Data Selection
   - Data Selection & Values

   If a menu option is italicized, then the option is unavailable for the report.

   After you set the options, the system might launch Work with Batch Versions so you can choose which version you want to run. Then, Version Prompting appears. Choose the prompting you want and click Submit to choose a printer and process the report.

See Also

- *Interactive Versions for Applications* in the *Foundation Guide* for more information about running J.D. Edwards ERP applications
- *Batch Versions for Reports* in the *Foundation Guide* for more information about running J.D. Edwards reports
- *Processing Options* in the *Foundation Guide* for more information about processing options and how to use them

Application Shortcuts

While you are working in the J.D. Edwards Web client, you can e-mail other users a shortcut to the application and form that you are looking at. The recipient double-clicks the shortcut in the e-mail to access your current position in the software.

Note

J.D. Edwards ERP allows you to send shortcuts that launch Windows client or Web client applications. Before you send a shortcut, you need to determine whether the recipient of the shortcut is using the Windows client or Web client version of J.D. Edwards ERP. However, you can only set up shortcut preferences on the Windows client machine.

► To send a shortcut to an application form

1. Launch a Web application and access the form that you want to send.
2. Click Tools and choose Send Shortcut.
3. On Send Shortcut, complete the following fields:
   - Address Number / User / Role / Distribution List

   Note
   If you are sending a shortcut to members of a distribution list, you must click the Distribution List option and then choose the address book number of the distribution list. If you enter the distribution-list address-book number without choosing the Distribution List option, the shortcut will be sent only to the distribution-list address-book number and not to the members of the distribution list.

   - Mail Box

     Choose which mailbox/queue you want the message to be sent to.

   - Subject

     Type the text that you want to appear in the Subject line of the e-mail message.
4. If you want to include a message with the shortcut, type it in the large field at the bottom of the form.
5. Click OK to send the shortcut.

   The recipient will receive the shortcut via an e-mail in the Work Center or a third-party e-mail system, depending on the recipient's e-mail preferences in ERP 8.

Web Application Forms

You interact with J.D. Edwards ERP applications through a series of forms. When you launch an application, its default form appears. The system displays other forms as required. Each form completely replaces the previous one, and you should use the buttons on the form to move through a series of forms instead of the back and forward buttons on your browser.

J.D. Edwards forms are composed of one or more of the following elements:

- Title bar
- Button
- Radio button/Checkbox
- Hyperlink
- Tab
- Field
- Detail area
- Tree control

Title Bar

The title bar appears across the top of the application. The title bar shows the name of the application on the left and contains three help buttons on the right. These buttons allow you to access application version information, online help for the form or application, and online help for individual form elements.

Buttons and Menus

Buttons reside on the both the toolbar and the title bar. Buttons might also appear in the main body of the form. When you click a button, the system performs an action such as closing a form, telling the system to process the information that you entered on the form, or launching another program.

An arrow in the lower-right corner of a button indicates that the button is a menu. When you click the button, the system displays the menu.

Usually, the system does not process the information that you have entered on a form until you click a button (such as OK or Submit).

Following is a list of the most common action buttons. Not all buttons appear on all forms.

- Select – After choosing a row in the detail area, click Select to perform an action particular to that record: open another form with more detailed information, launch a report, and so forth.
- Find – After completing one or more fields on the form with search criteria, click Find to fill in the detail area with the search results.
- Add – Click Add to add a new row to the table. Typically, a new form appears to facilitate your adding the record.
- Delete – Choose a record in the detail area to delete, and then click Delete to remove it from the table.
- Copy – Choose a record in the detail area to copy and then click Copy to create a new record based on the one that you chose.
- OK – After completing the fields on a form, click OK to process the information.
- Cancel – Click Cancel to close a form. If you click Cancel instead of another action button, such as OK or Find, the system disregards any data that you entered on the form.
- Close – Click Close to close a form. If you are on the default form for an application, clicking Close closes the application.

Following is a list of the most common menus. Not all menus appear on all forms.

- Row – This menu contains options that you can apply to a record in a detail area that you have selected.
- Form – This menu contains jumps to other forms within the application as well as form-specific options such as viewing attachments to the form.
- Tools – This menu contains options that are standard to most J.D. Edwards applications such user options, the ability to submit jobs and reports, the ability to export the contents of a detail area to a spreadsheet, and so forth.

Field and Detail Area

You enter data in fields and detail areas, and the system can display data in fields and detail areas. A detail area looks like a table and displays information with a series of columns. Each row represents a different record. Fields, on the other hand, display only one piece of data at a time. A disabled (grayed-out) field does not allow you to change the data that it displays. Additionally, some fields have a down-pointing arrow in them. When you click the arrow, you see a range of options from which to choose.

Sometimes when you click in a field, a button appears to the right of it. The button might look like a flashlight, a calculator, or a calendar. This button is called a visual assist. If you click the visual assist, the system helps you find and enter a valid value for the field. The flashlight brings up a form on which you can search for system information. For example, when you log in to the system, you might have several environments from which to choose. If you click the visual assist and choose an environment from the list, you do not need to worry about entering a valid environment and spelling it correctly. The calculator visual assist allows you to calculate a value and then enters the result in the field. The calendar visual assist allows you to pick a date visually from a calendar and then enters it into the field.

You might not need to enter a value in every field on a form. For example, many forms allow you to search for information, and they use the data that you enter in the fields to narrow the search. If you do not want to narrow the search based on a specific field, then enter a * in the field. A * is a wildcard character that tells the system that all values for that field are valid.

Tree Control

A tree control looks like two screens. One side displays a hierarchical structure of objects such as files, applications, and so forth. The contents of the other side change depending on the object in the hierarchy that you choose.

▶ To get help on a form

All help buttons appear in the upper-right corner of J.D. Edwards Web forms. Hover over the buttons to see their names.

1. To see information about a form such as its ID, the application it belongs to, and the software version, click About.
2. To see general information about how to use the current application, click Help.
3. To see information about a specific field, click Item Help.

   A question mark appears next to your cursor.

4. Click on the field for information about it.

   You can click on any number of fields.

5. When finished, click Item Help again.

   The question mark next to your cursor disappears, and you can continue using the form normally.

Dragging and Dropping in Web Application Forms

A parent/child form provides the ability to rearrange items in a form using the drag-and-drop feature. For example, in the Business Unit Structure Tree View program (P0006A), you can revise an organizational structure by dragging and dropping business units into a different level of the organization.

► To drag and drop items in a Web parent/child form

1. In a parent/child form, such as the Business Unit Structure Tree View program (P0006A), click the node of the folder or item that you want to move.
2. Click the Drag button.

   The system places a cursor icon next to the node that you want to move.
3. Click the node of the folder or item in which you want to place the drag item.
4. Click the Move button.

   The system places the drag item in the destination folder.

See Also

- *Parent/Child Form* in the *Foundation Guide* for more information about the parent/child form features

Detail Areas

Many forms in J.D. Edwards software contain detail areas. Detail areas display data. Depending on the application, you might be able to add, change, or delete data in the detail area as well.

When you first launch an application with a detail area, the detail area is empty. You must perform a search to fill the detail area. To perform a search, click Find. In many cases, if you click Find without providing any search criteria, the system assumes that you want to see all of the data in the underlying business view, and it displays the first few rows. Sometimes the system does not allow you to perform a search without first specifying some criterion.

J.D. Edwards provides a variety of methods for defining search criteria. Most detail areas have a row of blank fields above the column headings. This row is called the QBE (Query By Example) row. By entering data in one or more of the fields in the QBE row before you click Find, you limit the search based on what you entered. For example, in an application that lists employee information, if you enter Abbot in the field over the Last Name column, the system returns only those employees whose last name is Abbot. You can limit searches even further by entering data for other columns. If the detail area in this example had a column for city, for instance, you could also enter a city name, and then the system would return only those employees whose last name is Abbot and who live in the indicated city.

In addition to the QBE row, many forms also include fields above the detail area. Enter data in one or more of these fields to limit the search as noted above. Additionally, some forms also include radio buttons or check boxes which can help you limit a search in a particular way. For example, if you were looking at sales orders, the form might contain a check box that allows you to exclude incomplete orders from the list.

In the QBE row and in fields, you can usually use a * as a wildcard character. To return to the initial example, if you wanted to view all employees whose last name started with the letter M, you would enter M* in the Last Name column. Furthermore, many fields include a visual assist: a flashlight, a calculator, or a calendar. Click the visual assist to help you find or calculate a valid value for the field.

In all cases, search criteria is additive. That is, if you enter search criteria in the QBE row and the fields above the detail area, and you select other options, the system returns values based on all of that criteria. If a field is inactive (grayed out), then you cannot limit the search based on it.

Depending on how your system administrator has configured it, the system might load only a few rows of data into the detail area at a time. If more data rows exist than currently appear, the system displays a row counter and up and down arrows at the top of the detail area. Use the up and down arrows to load more data or to return to a previously loaded detail area set. The counter tells you where in the list you are. For example, when you first load a detail area, the counter might say Records 1–10. When you click the down arrow, the system loads the next ten rows and the counter says: Records 11–20.

To work with a specific row in the detail area, click the radio button or check box to the left of the row. The system highlights the row in the detail area to show that you have selected it. Then click Row to view the list of options for working with that data, including viewing any attachments to the row. If the detail area uses check boxes instead of radio buttons, you can select several rows at once. However, when you select multiple rows, you might not be able to choose the same options in the Row menu as you do when you select only one row.

Some detail areas only display data. Others allow you to enter data, but with the aid of a separate form. However, smaller detail areas might allow you to enter data directly into the detail area. In this last case, you can load data into the detail area manually or by importing the contents of a Microsoft Excel spreadsheet. To do so, the range that you specify in the spreadsheet should exactly match the columns in the detail area. Most detail areas also allow you to export their contents to a Microsoft Excel or Word file as well.

Finally, you can customize how a detail area looks. You can maximize the detail area so that it takes up most of your screen. You can change a column's color, and you can apply color and formatting to a column's text. You can define which columns to display in what order, which columns to sort on, and how wide a column should be. You can create multiple detail area formats for a single form so that you can view the data in different ways.

► To export detail area contents to Microsoft Excel or Word

1. Launch an application with a detail area, and then use Find to load the detail area with records.
2. Perform one of the following tasks:
   - To export the detail area contents to Excel, click Tools and select Export To Excel.
   - To export the detail area contents to Word, click Tools and select Export To Word.
3. Using the Export Assistant, specify the range of data that you want to export by clicking the first and then the last cell in the range.

   For example, if the detail area has four columns and three rows of data, but you want to export only the first three columns and the first two rows, you click the first cell in the first row and then the third cell in the second row.

Use the scroll bars in the detail area to bring cells into view. The system loads only a few rows at a time. Use the up and down arrows at the top of the detail area to load more rows or to return to a previously loaded row set.

Click Reset Selection if you clicked in the wrong cell.

4. Click Continue.

   The system exports the detail area contents that you selected to the appropriate file type and displays it.

► To import a Microsoft Excel spreadsheet into a detail area

1. Launch an application with a detail area that allows you to enter data directly.
2. Click Tools and select Import from Excel.
3. On Import from Excel Assistant, click Browse and browse to the Excel spreadsheet that you want to import.
4. Complete the following fields:
   - Worksheet to import from

     Enter the name of the worksheet containing the data that you want to import.

   - Starting Cell Col

     Enter the spreadsheet column letter containing the first cell in the range of data that you want to import.

   - Starting Cell Row

     Enter the spreadsheet row number containing the first cell in the range of data that you want to import.

   - Ending Cell Col

     Enter the spreadsheet column letter containing the last cell in the range of data that you want to import.

   - Ending Cell Row

     Enter the spreadsheet row number containing the last cell in the range of data that you want to import.

5. Click Import.

   The system imports the data that you selected.

► To create a grid format

If you want to recall the default format, save the original grid format before you save a new format. Otherwise, you must remove the new format, exit the application, and then access the application again to view the default grid format.

1. Launch the application for which you want to create a new grid format and click Customize Grid.
2. On Select Grid Format, click Create.
3. Enter a name for the format in Grid Format Name.

4. Complete the rest of the options as desired, and click OK.
5. To use the grid format as the default format for pervasive devices, click the format that you created and choose Default for Mobile Device.
6. Click Close.

► To create a grid format for pervasive devices

1. Using the Web client, launch the application for which you want to create a grid format for pervasive devices.
2. Click Customize Grid and choose the format that you want to use for pervasive devices.

   If you haven't created the format, do so now. Follow the same steps for creating the format that you use for creating a Web-based format. However, keep in mind the limited space and color options offered by most pervasive devices when you decide about columns to display, column widths, column and text colors, and so forth.

3. Click Default for Mobile Device and click Close.

► To apply a grid format

On any form with a grid, choose a grid format from the drop-down list next to Customize Grid.

You must create one or more grid formats before you can apply a grid format.

► To change a grid format

1. Launch the application containing the grid format that you want to change and click Customize Grid.
2. On Select Grid Format, choose the grid format that you want to change and click Modify.
3. To change the name of the grid format, enter a new name in the Grid Format Name field.
4. Change other elements of the grid as desired and click OK.
5. Click Close.

► To delete a grid format

1. On any form with a grid, click Customize Grid.
2. On Select Grid Format, choose the grid format that you want to delete and click Delete.

   The grid format disappears from the list.

3. Click Close.

► To hide or show grid columns

1. Launch the application containing the grid that you want to change.
2. Click Customize Grid and either create a new format or select an existing one to modify.

3. On Customize Grid, scroll to the Display and Order section.
4. To prevent a column from showing on the grid, choose it in the Display and Order list, and then click the left arrow.
5. To make a column appear on the grid, choose it in the Available Columns list, and then click the right arrow.
6. Use the up and down arrows to change the order in which the system displays the columns on the grid.
7. When finished, click OK, and then click Close.

► To rearrange grid columns

1. Launch the application containing the grid that you want to change.
2. Click Customize Grid and either create a new format or select an existing one to modify.
3. On Customize Grid, scroll to the Display and Order section.
4. In the Display and Order list, click a column name and use the up and down arrows to move it up or down in the list.

   The system displays the columns in the list in the order in which they appear from top to bottom. In other words, the column at the top of the list appears first on the grid, the column second from the top appears second on the grid, and so forth.

5. Repeat step 4 for any other columns that you want to move.
6. When finished, click OK, and then click Close.

► To set grid color and font

1. Launch the application containing the grid that you want to change.
2. Click Customize Grid and either create a new format or select an existing one to modify.
3. On Customize Grid, scroll to the Display and Order section.
4. Click a column name in the Display and Order list.

   The column name appears in the Selected Column field.

5. To apply a background color to the column, click a color in the pallet under Column Color.

   The hexadecimal value for the color that you chose appears in the Column Color field.

6. To apply a color to the text in the column, click a color in the pallet under Text Color.

   The hexadecimal value for the color that you chose appears in the Text Color field.

7. To apply a font style such as bold or italics to the text in the column, click the styles that you want to apply in the Text Options list.
8. Click Update Style.

   The system updates the Selected Column field to show you how your choices will look. The system also places a plus sign next to the column name in the Display and

Order list. This symbol indicates that user-defined formatting will be applied to the column.

9. Repeat steps 4–8 to apply formatting to additional columns in the grid.
10. When finished, click OK, and then click Close.

► To change grid column width

1. Launch the application containing the grid that you want to change.
2. Click Customize Grid and either create a new format or select an existing one to modify.
3. On Customize Grid, scroll to the Display and Order section.
4. Click a column name in the Display and Order list.

   The column name appears in the Selected Column field.

5. Enter a percentage value in the % Column Width field.

   This value is the percentage of the space that you want the system to allot to the column based on the width defined for the data item on which the column is based. You can enter a value between 25 and 400.

6. When finished, click OK, and then click Close.

► To change the sort sequence of a grid

1. Launch the application containing the grid that you want to change.
2. Click Customize Grid and either create a new format or select an existing one to modify.
3. On Customize Grid, scroll to the Data Sequencing section.
4. If you want to sort on a column, click the column name in the Available Columns list, and then click the right arrow.

   The system moves the column name from the Available Columns list to the Sequenced Columns list.

   A column must be included in the grid—that is, its name must appear in the Display and Order list—before you can sort on it.

5. If you do not want to sort on a column, click the column name in the Sequenced Columns list, and then click the left arrow.

   The system moves the column name from the Sequenced Columns list to the Available Columns list.

6. To rearrange the order of sort precedence, use the up and down arrows under the Sequenced Columns list to rearrange the column names.

   The system first sorts by the column at the top of the list, then by the column second from the top, and so forth.

7. To sort column values in ascending order, click the column name in the Sequenced Columns list and click Ascending.

If you leave the Ascending box blank, the system sorts the column in descending order.

An A appears next to the column names to be sorted in ascending order, and a D appears next to the column names to be sorted in descending order.

8. When finished, click OK, and then click Close.

Media Object Attachments

J.D. Edwards media object and imaging features allow you to attach information to a program, including information that might currently exist as a paper-based document. For example, you can use a text attachment to explain special circumstances regarding a journal entry. The media object feature allows you to attach the information to J.D. Edwards software programs, forms and rows, and Object Librarian objects. The imaging feature, within Media Objects, provides flexibility for creating a more efficient method of information storage.

Use Media Objects to link information to programs, either to individual rows in a grid or to a form. The following list describes the types of information that you can attach to a detail area row or to a form:

Text

Media Objects provides a word processor that lets you create a text-only attachment. For example, you could use a text attachment to provide specific instructions for a form or additional information about a record.

Image

Images include files such as Windows bitmaps, GIF, and JPG files. These files might represent electronically created files as well as scanned images of paper-based documents. For an image to be available to be attached, your system administrator must first add it to an image queue.

Object Linking and Embedding (OLE)

Media objects can be files that conform to the OLE standard. OLE allows you to create links between different programs. Using these links, you can create and edit an object from one program in a different program. J.D. Edwards software provides the links that you need to attach OLE objects.

You attach OLE media objects at the base form level. Media objects attached at this level are attached to a form and not to any data that might appear on the form. You can attach media objects to a grid row or a form, but the files themselves exist in separate directories. The only file information included with the program to which the OLE links is the path to the supporting file.

You can use only OLE objects that you properly register and install as OLE objects through Windows.

Uniform Resource Locators (URL)/Files

Media objects can be links to web page URLs or other related files. When a developer attaches a URL media object to a control object on a form, the web page appears as part of the form. When a user attaches a URL to a form or Object Librarian object, the media object acts as a link to the URL. Files can reside in an image queue, or you can attach a local or networked file.

When you attach a media object to a form, the attachment might not be available if you access different data on the form. For example, if you attach a media object to a form that contains data for order number 2002, this attachment does not appear on the form that appears when you access data for order number 3003. The base form is the same for both order numbers, but the data associated with the form is specific to each order number. The order number represents the key to the location where an attachment is stored.

If attachments exist for a form, a paper clip icon appears at the right of the status bar when you open the form. For an OLE object attached at the base form level, a document icon appears at the right of the status bar.

When you first load a grid, grid rows do not indicate whether attachments exist for the corresponding records. To determine whether attachments exist for any of the records, you must search for the attachments. The system searches for attachments only on records currently loaded into the grid. When you click the Find button to refresh the records in the grid or to display new records, the form resets the attachment's view status. You must search again to display the attachments for the grid records.

You use the Media Object Viewer to examine media objects already attached to an object and to attach new media objects to it. The Media Object Viewer workspace is split into two panels. The left panel is the icon panel and the right panel is the viewer panel. Icons for any files previously attached to the record appear in the icon panel. To view an attachment, you click its icon, and the system displays its contents in the viewer panel. An object can have multiple attachments.

You can use templates to create a format for a frequently used media object. A template might include attachments of its own, such as images. For example, you can create a letterhead and a standard form for a memo. You system administrator creates templates.

See Also

- *Media Object Attachments* in the *J.D. Edwards Web Client Guide* for information on how to attach media objects in the Web client
- *Working with Templates* in the *Foundation Guide* for information about how to create text templates
- *Processing Media Objects* in the *Form Design Aid Guide* for information about how a software developer sets up a form to handle media objects
- *Media Objects and Imaging* in the *System Administration Guide* for information about how a system administrator configures J.D. Edwards ERP software to enable media objects

► To view attachments

1. Launch an application with a grid and click Find to fill the grid with records.
2. Click the button with a magnifying glass and a paperclip on it (the button is left of the column headings).

   The system marks the rows that have attachments by displaying a paperclip on the buttons on the left of the grid.

   Each time you refresh, perform a search, or move back to view a previous record set, you must search for attachments again.
3. Click a button with a paperclip icon to view the attachments for that row.

4. On Media Object Viewer, click the object in the left column corresponding to the media object that you want to view.
5. When finished, click Cancel.

► To attach text

1. Launch an application with a detail area and click Find.
2. For the row to which you want to add an attachment, click the button left of the grid.
3. On Media Object Viewer, click Text.
4. In the viewer panel, type the desired text.

   You can use the formatting tools at the top of the viewer panel to format your text.

5. Click Back to return to the application.

► To attach text using a template

1. Launch an application with a detail area and click Find.
2. For the row to which you want to add an attachment, click the button left of the grid.
3. On Media Object Viewer, click Text.

   The system creates a new, blank text object.

4. Click Templates.
5. On Work With Media Object Templates, click Find to load the grid.
6. Find the template that you want to use and click Select.

   Media Object Viewer appears with the template contents pasted into the viewer panel.

7. In the viewer panel, type the desired text.

   You can use the formatting tools at the top of the viewer panel to format the text of your note.

8. To return to the application, click Back.

► To attach an image

1. Launch an application with a detail area and click Find.
2. For the row to which you want to add an attachment, click the button left of the grid.
3. On Media Object Viewer, click Image.
4. Choose an image queue from the drop-down list.
5. Click the image that you want to attach.

   To preview images, click Preview and then click the image.

6. Click Add.
7. To return to the application, click Back.

► To attach a URL or file

1. Launch an application with a detail area and click Find.
2. For the row to which you want to add an attachment, click the button left of the grid.
3. On Media Object Viewer, click URL/File.
4. To attach a file residing in an image queue, perform the following steps:
   a. Click Select Queue.
   b. Choose an image queue from the drop-down list.
   c. Click the file that you want to attach.

      To preview files, click Preview and then click the file.
5. To attach a file residing in another location or a URL, perform the following steps:
   a. Click Select URL/File.
   b. Enter the URL or File name and location or click Browse to browse for the file.
   c. To preview the file, click Preview URL/File.
6. Click Add.
7. To return to the application, click Back.

► To attach an OLE object

1. Launch an application with a detail area and click Find.
2. For the row to which you want to add an attachment, click the button left of the grid.
3. On Media Object Viewer, click OLE.
4. On Insert Object, to create a new object, choose an object type, and then click OK.

   Selections vary from system to system depending on what the system administrator installs on your workstation and on the network.
5. Create your object.
6. To attach an existing object, choose Create from File, locate the object on your system, and then click OK.

   Depending on whether you create an object or attach a preexisting object, the application associated with the object appears in the viewer panel to display either a blank workspace or the preexisting object.

   The menu bar displays the menus for the application from which you call the object. For example, if you select an Excel document, the menus for Excel appear on the menu bar.
7. To return to the application, click Back.

► To remove an attachment

Note

When you remove an attachment from an object, you break the connection between the object and the media object. The media object itself is not deleted as a result of your removing its attachment to an object.

1. Launch an application with a grid and click Find to fill the grid with records.
2. Click the button to the left of the column headings with a magnifying glass and a paperclip on it.

   The system marks those rows that have attachments by displaying a paperclip on the buttons on the left of the grid.

   Each time you refresh, perform another search, or move back to view a previous record set, you must search for attachments again.
3. Click a button with a paperclip icon to view the attachments for that row.
4. On Media Object Viewer, click the object in the left column corresponding to the media object that you want to delete.
5. Click Delete.

   The icon for the media object disappears.
6. When finished, click Cancel.

Designing Web Applications

This section describes how to design and produce J.D. Edwards software applications for use on a Web client with the J.D. Edwards Toolset. It provides information about similarities and differences between designing for HTML clients and designing for Windows clients. It also provides valuable best-practices guidelines, tools, and techniques that will help you design more usable J.D. Edwards Web applications. This section is intended for application developers.

The guidelines in this section are suitable for J.D. Edwards employees, J.D. Edwards business partners, and J.D. Edwards clients.

See Also

- *J.D. Edwards Development Tools Guide*
- *J.D. Edwards Development Standards: Application Design Guide*
- *J.D. Edwards Portal Guide*

User-Centered Design Guidelines

To create usable designs, you need to follow a user-centered design (UCD) technique. User-centered design means that you get early and frequent user interaction with the real user community to get feedback and input into the design of the application. Before beginning application development, you must clearly document and validate the goals and business objectives with users through conceptual and cognitive design reviews. All team members should understand who the application users are and what goals the users expect to achieve with the application.

User Analysis Checklist

- Define the users
- Define user and business goals
- Define user tasks
- Create use cases
- Perform usability evaluations

Define the Users

You must understand your audience before you can design an application that will enhance the user experience. Therefore, you must develop a user profile for each user type (end user, power user, and so forth) that you expect to use the application. You can gather information for a user profile from various sources such as Web surveys, interviews, task analyses, contextual inquiries, focus groups, and market and competitive analyses. The following questions are the key questions that you should ask:

- Who are the targeted users? (For example, B2B e-collaborative suppliers and buyers)
- What are the targeted users' job titles? (For example, e-Procurement Managers, Buyers)

- What are the targeted users' levels of expertise? (For example, novice Web users, experts in the supply chain business)
- What are their unique needs? (For example, do they need to use this application remotely, when they are mobile, and so forth)
- What types of computers do the targeted users use? (For example, desktops, handheld devices, screen resolutions and sizes, Web browsers)
- What is the targeted users' work environment like? (For example, shop floor, cubicle, office)

Define User and Business Goals

You must understand how users move from goals to tasks to actions. Some user goals might be as simple as "doing time entry quickly and accurately." The business goals might be to increase revenue or decrease the cost of providing support. Successful products are designed by understanding both user and business goals. You can define user goals through contextual inquiries, user interviews, and observations.

The following questions are the key questions that you should ask:

- What is the user's personal goal in using the applications? For example: Doing data entry as quickly as possible to meet incentive measures.
- What are the key goals for the targeted users to accomplish?
- What are the overall business goals? For example: To increase daily sales order entry.

Solutions for Different Users

In any given enterprise, only a fraction of the employees have access to the enterprise application suite. Most enterprises would like to achieve a better return on the technology investment and reap the benefits of consistent communication across the organization. However, they would like to meet these goals without equipping every desktop in the organization with a fully loaded desktop machine. Corporate intranets have achieved widespread popularity due, in part, to the low-cost, streamlined distribution of information that they provide. Outside of the constraints of traditional paper-based publishing, companies find they can dramatically improve the information flow within the organization.

However, most intranet-based information is relatively static. In contrast, information within the enterprise system tends to change with each new transaction. Companies would like to marry the ease of distribution afforded by the corporate Intranet with the real-time accuracy and processing capabilities of enterprise applications.

J.D. Edwards software allows businesses to leverage the corporate intranet to increase access to enterprise applications. Any browser-equipped device can provide a real-time window into the enterprise's information resources. Because of the low technology overhead, businesses can include more users in the information flow, thus achieving the following benefits:

- Quicker dissemination of information to a broader corporate audience
- The bringing together of disparate business operations and distributed sites
- Better communication of goals, priorities, and strategies

- Improved decision making through the increased availability of information at every organizational level

The User Spectrum

Not every user in an organization uses enterprise applications in the same way. Some need continual access and the full suite of capabilities and desktop tools afforded by a robust client environment. Others require only the ability to review statuses and enter straightforward transactions. The spectrum of needs presupposes different technology requirements throughout the enterprise, depending on the user's skills and job requirements.

User requirements from enterprise applications vary depending on their role in the organization. Analytic users leverage multiple desktop tools to interpret and package enterprise information. Action users rely on quickly available, easily accessed information.

Analytic Users

At one end of the spectrum are users who gather, analyze, repackage, and distribute information to the rest of the enterprise—the knowledge workers of the organization. These users rely on a range of desktop tools, including enterprise applications, spreadsheets, and publishing tools, to bring together the various islands of information in the organization and integrate them into a meaningful whole.

J.D. Edwards client and server modes provide these users with the OLE-based, fully integrated desktop they need to maintain the information flow within the organization. The Windows client unites the processing tools of personal productivity applications and enterprise applications, as well as their respective data resources. Action users can then integrate this information, analyze it from various departmental perspectives, and repackage it in a context relevant to multiple functional areas, for example, requirements planning, executive decision making, marketing, purchasing, and so on.

Action Users

At the other end of the spectrum are those users who review information so that they can perform a particular action. This class of users crosses organizational levels and, for example, might include order entry clerks, shop floor personnel, and executives. Action users often use the system to review order status and item availability, for example, but they do not repackage and publish information for subsequent distribution throughout the enterprise. In browser mode, J.D. Edwards software offers action users the access they need with a point-and-click interface that minimizes training. At the same time, the low-overhead client enables the business to extend access to those action users previously out of the information flow due to the cost of equipping them with a fully loaded client workstation. The enterprise can push applications out to these users over a standard TCP/IP network to any browser-equipped device. Because no J.D. Edwards code resides on the client, the business also gains the benefits of centralized software maintenance and upgrades for an entire class of users.

Enterprise Applications Considerations

Like users, enterprise applications tend to fit better with either client/server or browser mode depending on the role that the application plays in the information flow. Applications that provide the tools for analyzing and manipulating information from a variety of sources are well served by client/server implementations. J.D. Edwards client/server modes furnish the needed interactivity and graphical support, as well as the advantage of dedicated use of the client's processor. Specific examples include:

- Modeling and prototyping
- Budgeting and forecasting

- High-volume transactions requiring custom interfaces by transaction or customer type

Applications intended to communicate and distribute shared information and to support standard transaction entries work well in browser mode. The types of applications listed below display information to good advantage without hindering the client's resources, making them ideal candidates for J.D. Edwards browser mode:

- Information gathering and presentation applications
- Inquiry-based self-service applications
- Repetitive standard transactions applications

Even with the complementary fit between computing mode and application environment, business needs often dictate that both modes be available to adapt to real-time changes in the business. In the J.D. Edwards environment, both modes are inherent in the architecture, to be deployed when and as needed. The enterprise can use a combination of both modes, maintaining consistency in business data and processes. By offering both client/server and Web-based access to enterprise applications, J.D. Edwards software can meet the full spectrum of user needs within the enterprise.

Because client/server and browser modes exist in a single software solution, the business can implement J.D. Edwards software to match user needs or the information requirements of a given business process. With more users accessing the enterprise's information resources, the business realizes significant benefits:

Tighter integration of distributed business units	More users throughout the organization have access to a single, consistent source of information.
Streamlined processes	Traditional paper-based processes are more easily automated.
More efficient decision cycles	Decisions are not always pushed to the limited number of users who have access to enterprise applications.

Despite the differences between the two modes, enterprises are moving to combine client/server and browser solutions in a single computing solution, such as that provided with J.D. Edwards software. Where and how the enterprise deploys each mode depends on the business need and the built-in flexibility of the solution.

In considering the combined use of Internet and client/server technology in the enterprise, businesses face two fundamental challenges:

- They must identify those areas of the business best served by Web-based solutions and those areas best served by client/server solutions.
- They must implement solutions that accommodate both client/server and browser modes.

Define User Tasks

A user task analysis defines all the tasks that the users perform with the system to achieve their goals. Performing a task analysis allows you to:

- Generate ideas for new products
- Identify essential features to include in products
- Design the user interface for products that are already identified and for which the scope has been determined
- Improve the usability of products already in production

To drive a user-centered design approach rather than a function-centered design approach requires an understanding of the user's tasks and the context in which those tasks will be performed. As with goals, a user task analysis can be gathered using contextual inquiries, interviews, and observations. The following questions are the essential questions that you should ask:

- What tasks do the users need to perform using the system?
- What are the critical and important aspects of their tasks?
- In what sequence do they perform those tasks? In other words, what is the workflow of the task?
- What are their current environmental constraints and issues?
- How can their current work processes be improved by using the system?

Task Analysis Example

This example uses the E-procurement self-service application as a model for creating a task-based, user-centered design. The following is an example of the task analysis:

- The customer accesses the application to see customer alerts on changing market needs.
- The supplier responds to the alerts in a timely manner.
- The supplier receives critical data about the buyer/seller market just in time, such as data about shortages.
- The supplier searches for the specific order request.
- The supplier browses the current schedule for shipment tracking.

The example below lists all of the tasks that a user might perform in the system as a customer or a supplier.

#	Customer Task	Supplier Task
1.	Change a user profile	Change a user profile
2.	Request a new user ID or password	Request a new user ID or password
3.	Monitor daily alerts from enterprise (first)	Monitor daily alerts from enterprise
4.	View prioritized schedules and work	View prioritized requests, such as rejected orders

5.	View daily and delayed delivery schedules	Notify of daily and delayed shipment schedules
6.	Send requests for quotes to a supplier	View requests for quotes from buyers
7.	View responses from suppliers	Respond to requests for quotes from buyers
8.	Search a status of a specific order or item ordered from a supplier	Search for the status of a specific order or item for a buyer
9.	Track outstanding shipment statuses	Track deliveries
10.	Request an unfulfilled order shipment	Respond to a shipment request
11.	Review outstanding invoices to be paid	Review overdue invoices
12.	Create sales orders	View sales orders
13.	View sales order status	Respond to sales order status
14.	Review supplier inventory	Review buyer inventory

Create Use Cases

A *use case* is a method for modeling user tasks. The purpose of use-case design is to model user tasks in flowcharts to understand the navigational structure behind the Web application design. A use case describes the tasks that the user wants the system to do, such as querying the status of an existing order. A use-case approach helps define the boundaries of the system and prevents the growth of scope that can often happen without a clear model of use cases.

In the process of identifying and defining the participants and use cases, the designers define the application scope, or what can be done and what cannot be done within the application.

Use cases provide the following benefits:

- They provide an easily understood communication mechanism.
- They reduce the risk that requirements will be overlooked (when requirements are traced).
- They provide a concise summary of what the system should do at an abstract (low modification cost) level.
- They use the language of the customer.

Usability Engineering

Designing J.D. Edwards software applications using user-centered design principles is not enough in and of itself. As application development progresses, you must validate and verify the designs with the end users. Usability engineering includes conducting iterative usability evaluations of the designs during the design cycle. Incorporating user-centered design and usability engineering into the development cycle provides the following benefits:

- Increases user efficiency and performance
- Increases operational efficiencies and productivity
- Improves user experience
- Improves customer satisfaction
- Increases sales and builds customer loyalty
- Reduces the overall development costs
- Reduces training and support costs

The case study shows how and why designers changed a form after performing usability evaluations.

In response to the evaluation, designers sought to redesign the form to make it more usable. They decided to make the following changes:

- Emphasize the J.D. Edwards branding
- Use graphical icons that are inoffensive to international audiences
- Incorporate more meaningful labels and headings
- Allow more user-defined layout of screen elements based on user task flow
- Reduce the amount of on-screen text by incorporating clear labels and headings
- Reduce clutter and visual noise
- Incorporate a better grouping of information

Information Architecture

The presentation of most of the J.D. Edwards Web applications is created using the Form Design Aid (FDA) tool.

Create Wire Frames

One technique used in Web application design is the process of creating wire-frame diagrams to describe the initial Web-screen design mock-up. A wire-frame diagram provides a placeholder for your screen content and serves as a high-level architectural blueprint for the Web application. Incorporating wire frames into the development cycle helps speed up the process of creating the Web application.

Use cases allow you to::

- Conceptualize the navigational flow from one screen to another
- Identify the contents of the Web application
- Provide an overview of the Web application
- Develop a conceptual prototype of the design
- Reveal problems in how tasks are distributed over the interaction spaces
- Provide a rough overview of how complex the system will be for users (overly long chains of transitions invite review for possible consolidation and simplification)
- Serve as a powerful tool for understanding the overall organization of software
- Reveal lurking problems in the existing designs and help clarify possible solutions

Basic Design Guidelines

Observe the following guidelines to design Web applications while working within the constraints of the FDA tool:

- Organize the information based on the user's task needs. The layout must represent the user's task flow.
- Display critical and frequently used information first.
- Avoid horizontal scrolling—users dislike scrolling horizontally.
- Use subheadings to group information into sections.
- Avoid using long, wordy sentences or phrases in instructions or labels. Use action verbs and active voice to describe the actions needed. The user interface should be self-evident and require minimal instructions.

Define the Navigation

Based on use cases and wire-frame diagrams, you define the navigation model of the Web application. Currently, the J.D. Edwards navigation scheme is sequential: users can move only in one direction by closing a form and returning to the previous form. However, Web applications require greater flexibility in designing the navigation scheme because they include the Back button. If the users cannot achieve their goals in the Web application, they are likely to click the Back button on the browser to exit an application.

Basic Navigation Guidelines

The following list presents 10 guidelines for designing a successful navigation for Web applications and Web sites. Navigation should:

- Be easy to learn
- Remain consistent across the application
- Provide feedback to users
- Appear in the user's context (for example, error messages)
- Offer alternative navigations
- Be efficient for the user
- Provide clear visual messages
- Provide clear and understandable labels
- Be appropriate for the application
- Support the user's goals and behavior

Navigation Schemes

One important challenge for Web designers is how to organize information in the Web forms. The navigation scheme must provide users quick and easy access to the information they need. Depending on the purpose of your design, you should use one of the following types of navigation schemes:

- Task-Based

  Navigation is based on the user's task flow. This scheme is the most effective for J.D. Edwards Web applications. The J.D. Edwards Task Explorer is an example of a task-based navigation. Task-based navigation follows a browse path, but should not be too deep. Users should not have to go through four to six levels of hierarchy to reach their specific tasks.

  J.D. Edwards Self-Service applications use a task-based navigation model. Users navigate through the Enterprise Portal to access the workspaces where they can then access specific tasks.

- Sequential

  The navigation is based on one step at a time and is linear in structure. Use this type of navigation for a dialog-driven, tutorial style design.

- Informational

  Nonlinear design by which users can jump and skip pages using a hypertext model and information is organized by category. The user does not have a specific path to follow and requires more flexibility in browsing various information. This type of navigation scheme is used in sites such as Yahoo!. The J.D. Edwards Portal workspace also uses an informational navigation scheme.

Links

If you are designing an application with an HTML appearance, such as a self-service application with a lot of customized HTML code, then you should use links as your navigation mechanism. If you are designing a Windows-based J.D. Edwards application with grids, tabs, and other J.D. Edwards controls, then it makes sense to use J.D. Edwards standard navigation schemes such as the J.D. Edwards toolbar and menu bar.

The J.D. Edwards software menu items such as Find, Select, OK, Cancel, and Close are familiar to the end user and offer standard run-time processing. You should use these standard menu items instead of customized buttons whenever possible.

Observe the following guidelines when designing navigational links in J.D. Edwards Web applications:

- Use action verbs to describe the task, for example, View Account Information.
- Avoid making users take a different path when completing a task, for example, Browse Catalog.
- Provide a clearly marked and easy-to-see Close button to allow users to save and exit the application.
- Use hyperlinks to connect users to additional information. In a J.D. Edwards software grid, use hyperlinks instead of the two-step Select action.
- Make hyperlinks easy to understand. Avoid leaving the user guessing about what information will appear. For example, avoid providing a Travel hyperlink that displays information about travel agencies.
- Use a single word or a short phrase for links. At the most, use two to three words.
- Avoid making links for long sentences or paragraphs.
- Avoid using an underline for labels that are not links. Make sure users can distinguish between a link and other words that are only being emphasized.
- Show *unvisited* links in blue with an underline. An exception is the Portal navigation bar, where there might be a conflict of color. In this case, use a contrasting color link. This contrasting color should be designed so that the link does not become invisible.

  Show *visited* links in purple with an underline.

  Show *currently selected* links in red where the cursor is positioned.

- Provide an e-mail link in blue and underlined with correct e-mail labels, for example, E-mail helpdesk.
- Avoid using “click here” for more information. Create the link on the most relevant word in the sentence.

Correct	Open Sales Orders
Incorrect	Open Sales Orders: Click here!

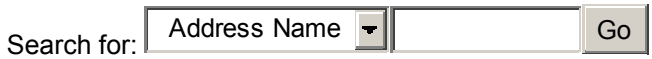
Define the Search Interface

For Web users, you must make the search function simple and fast to perform. The search function is one of the most important user interface elements for Web applications. Usability studies indicate that only 64 percent of users find what they need on the Web because of poorly designed search applications. Use the guidelines in the following topics to design searches.

Simple Search

A simple search is required for the novice user who knows what he or she is looking for. It must be easy to use. Typically, simple search functions include minimal filtering options. In J.D. Edwards Web applications, a simple search function should be provided in the Portal home page or in each workspace. Observe the following guidelines to design a simple search:

- The area to search should be clear, for example, J.D. Edwards Database or the World Wide Web.
- The search functions should be available from the home page or from the Portal.
- A simple search should not take the user to another page using a hyperlink.
- Provide a link to the Advanced Search function from Simple Search.

  Advanced Search

  If the component or screen space is limited, move the words “Search for:” above the control. For example:

  Search for:

  Advanced Search

- Use the Go button label when space is limited, such as in a Portal component. Otherwise, use the Search button label.
- Map the Go button to the <Enter> key.

Advanced Search

The Advanced Search function is used to find specific information. Users can use the Advanced Search function to narrow the scope of their search by specifying one or more fields of information. Observe the following guidelines when designing an advanced search:

- For expert users, provide an advanced search option that replaces the current Query by Example (QBE) search function in J.D. Edwards software.
- Provide a hyperlink for the Advanced Search function on main forms and on the Simple Search form.
- Use the search default <contains> to replace the QBE function.
- Provide users with a way to navigate to the simple search from the Advanced Search page.
- Logically group the search criteria and clearly indicate the different options for searching.
- Allow users to refine the search on the same page that displays the results.
- Use Arial as the font for the text-entry text box because it is a narrow font and allows users to enter more characters.

Displaying Search Results

Observe the following guidelines to display search results:

- Display search results on the same page from which the search was performed.
- Allow users to refine the search further if required; add a Refresh button for a refined search.
- The title of the search results section should be clearly labeled **Results** in bold.
- The search results should be shown in batches of 10 records.
- The Search function should allow users to return to a specific section displaying.
- Provide “bread-crumb” navigation back to the visited pages or links, for example, 1 2 3 4.
- Provide <Next> and <Previous> links rather than <Forward> and <Back>.
- If no records are found, display a clearly visible *No Records found* message, for example, in the top left corner.

Search Error Recovery

Observe the following guidelines to facilitate the user’s ability to refine searches:

- Display the error message above the same form on which the user started the search so that the user can refine the search or correct the search parameters.
- Make all error messages clear, constructive, and specific.
- Allow users to begin a new search if no records are found.
- Offer the user tips to refine the search further.

Functional Differences between HTML and Windows

J.D. Edwards software offers three distinct interactive client experiences:

- Microsoft Windows Full Client
- Windows Terminal Server Client
- HTML Client

You can mix and match these clients in a J.D. Edwards software implementation to support a full range of enterprise-wide solutions. The common component that is used by all client solutions is the J.D. Edwards software specification. Because all client solutions access the same J.D. Edwards software specifications, developers can employ the Write Once, Run Everywhere strategy for all J.D. Edwards applications and reports.

Nevertheless, an HTML client functions somewhat differently from a Windows client. Anyone using the Form Design tool to produce HTML applications alone or in addition to Windows applications must understand these differences so that the same applications can be designed to function efficiently in both environments.

To create Web applications, you use the J.D. Edwards Development Toolset in the same manner as you would for Windows applications. To generate the application for the Web:

1. Save the application in Form Design Aid.
2. Generate the application to the Web server.
3. Attach the application to a menu and designate in which mode the application should run.
4. Execute the application.

Currently, the process of generating Web applications is relatively manual. J.D. Edwards intends to automate parts of this process in the future.

Appearance Differences

The Windows client runs as a Windows application. It uses windows, frames, dialog boxes, menus, toolbars, and other GUI components provided by the windows platform.

Conversely, the HTML client runs in a Web browser. It uses the common controls provided by the HTML standards implemented by the browser. These common controls appear and feel different than the windows GUI components.

In the future, HTML rendering will use absolute positioning most of the time. Consequently, the system will render a control at the exact position as it is defined in Form Design Aid (FDA) and with the exact size as designed in FDA. Because the Browser and Microsoft Windows Application display fonts differently, the same font might appear bigger or smaller in one platform than in the other. It is the application developer's responsibility to ensure that a control is big enough for both Windows and HTML clients.

The exception to the absolute positioning rule is the rendering of grid controls. To better use the browser's screen space, the HTML client usually renders the grid to be as wide as the form. In other words, the right boundary of the grid is stretched to the right boundary of the form. Controls to the right of the grid are rendered below the grid. Controls to the left side of the grid are rendered to the left.

J.D. Edwards recommends that you do not put controls to the left or right of the grid control because the grid is usually the operation in the center of the form.

Behavior Differences

The Windows client is field-based; for example, when the user uses the Tab key to move out of a field, the data-validation routine is run immediately. The user immediately sees the associated description and the error status of the data field.

Conversely, an HTML client is page-based; for example, when the user tabs out of a field, that piece of data typically is not processed immediately. The events for that field are not launched immediately, either. Rather, these events are queued. When the whole page is sent to the JAS server, all the queued events are run on the server and all the controls are processed in a fashion similar to the Windows client control processing. The process of sending the page to the server is called a post.

After the JAS server receives the page with input from the user, it processes all input in the sequence in which it was entered. During this process, the enterprise server is used to process business functions, carry out database queries, and so forth. In our example, the data validation routines will be processed and the associated description will be loaded from the enterprise server. Then the processed page is sent back to the browser. This whole process, from the post to the return to the browser, is called a round trip. At this point, the user can view the associated descriptions of all the data fields and the error status of each.

How the JAS Server Processes a Post

The HTML Client runtime engine maintains virtual images of the open forms in the system. When the JAS server receives a new page posted by the browser, it first finds the correct virtual client image for that user session. Then it processes all data on the form in the order shown below. The following sequence does not necessarily include all server processing; rather, it outlines the sequence of events.

1. Execute the Control is the entered event.
2. If the control has changed, copy data sent from the browser into form controls of the virtual client that is running on the JAS server.
3. Execute the following events if a control value has been changed:
    - Control is exited
    - Control is exited and changed inline
    - Control is exited and changed asynchronously
4. Validate and format the control.
5. Repeat steps 1 through 4 for all controls that have been changed.
6. Execute the following events and runtime logic if a QBE or grid cell value has been changed:
    - Column is entered
    - Column is exited
    - Column is exited and changed inline
    - Column is exited and changed asynchronously
    - The grid cell is validated and formatted

- Row is entered
- Row is exited
- Row is exited and changed inline
- Row is exited and changed asynchronously

7. Repeat step 6 for all grid cells and rows that have been changed.
8. Execute Button Clicked events, such as OK Button Clicked, that caused this page to be posted.

Actions That Trigger a Post and Input Processing

The following user actions will trigger the browser to post the Web server:

- Clicking a toolbar button such as OK or Cancel
- Clicking a form exit, row exit, or tools exit
- Clicking a radio button or check box that has event rules in the Selection Changed event
- Clicking a push button or bitmap
- Clicking a hyperlink such as clickable static text, a clickable grid cell, or clickable text block control
- Clicking visual assist for a text field or a grid cell
- Clicking the paper clip image of a grid row
- Changing tab pages
- Clicking the Previous/Next link for the grid
- Clicking the edit button on single line grid
- Clicking an editable grid for the first time
- Performing actions on a tree control or the tree in parent/child control such as:
  - Expanding a tree node
  - Collapsing a tree node
  - Dragging and dropping a tree node
  - Double-clicking a leaf node if it has event rules in the Double-Click Leaf Node event.
- Clicking the Tools/Refresh menu

The input processor manages input that triggers a post. These input processor events are separate from ER events; in fact, input process events do not necessarily require that ER events be attached. The input processor resolves posted inputs in a specific order, and it looks for specific cues to determine what changes occurred. The input processor processes events as it detects them; consequently, some programmatic changes to the grid can be interpreted as if the user had entered changes. The following list shows the order in which the input processor manages events, how it detects changes, and what actions it triggers:

1. If the Shortcut or Back button was clicked, the processor performs that action and exits (it performs no further processing).
2. The processor processes QBE events, if appropriate.
3. With the exception of the grid, PC events, and the tree, the processor processes the control events that the user visited or accessed. The system uses JavaScript to detect these conditions.
    - Text fields – The processor sends "entered" and "exited" events to all controls that the user visited. It also sends "changed" events to the controls that the user accessed.
    - Check box – The processor looks for a changed state and sends the "changed" event if appropriate.
    - Radio buttons – The processor sends a “clicked” event only (the control itself detects changes).
    - Combo box – The processor reviews the selection and sends an event if appropriate.
4. The processor processes grid events or parent-child events.
    - Grid events – JavaScript does not indicate where the user has visited, so it sends the entire contents of the grid. The processor compares the previous values in the grid to the current values. If any have changed, it sends a "cell edit" event to the grid. This event launches grid focus, row entered, column entered, and later column exited/change and row exited/changed events. When processed, these events occur in sequence, top-to-bottom and left-to-right. The system uses this same logic for silent posts, except that only a few rows are sent at a time.

      The processor also selects rows that the user has turned on using a check box.

    - Parent Child events – The system can send one of these mutually exclusive events for the PC control:
        - Cancel drag drop (drag was started, but next action was not drop)
        - Row selections
        - Next/Prev page
        - Expand/Collapse node
        - Drag
        - Move
        - Copy
5. Process tree events – The system can send one of the following mutually exclusive events for the tree control:
    - Row selections
    - Next/Prev page
    - Expand/Collapse node
    - Double-click node

6. Process click events – The system can send one of the following mutually exclusive events:
    - Customize Grid
    - Export/Import
    - Refresh
    - Load Form (opening from link or menu)
    - Grid Prev/Next page
    - Form/Row/Report/View Menu Exit
    - Clicked Hyper Link column on grid
    - Tab Page Clicked
    - Grid Tab Clicked
    - Grid Next Line (single line edit)
    - QBE/Cell Visual Assist
    - Row double-clicked (select)
    - Row Header double-clicked (MO)
    - Embedded MO event
    - Check attachments
    - Textblock hyperlink clicked
7. Button clicked – These events include all the standard buttons.

Actions That Do Not Trigger a Post

The following events will not trigger a post automatically and, therefore, will not result in a round trip:

- Using the Tab key to move out of a text field
- Using the tab key to move out of a grid cell
- Using the tab key to move out of a QBE cell
- Selecting a tree node in a tree

Managing Round Trips

A round trip has an adverse impact on performance because it uses network resources. Too many round trips reduce the scalability and performance of the system. A round trip also has an adverse impact on user performance. When a round trip occurs, the user must wait for the server to process the page and return it to the browser. When the page does come back, it flashes on the browser. Although round trips can have a negative impact on performance, they are acceptable (even expected) when data accuracy is critical.

Because of these impacts, an important design principle for Web-based applications is to avoid unnecessary round trips. However, after carefully considering the performance impact of the post and eliminating alternative solutions, if an application still requires some events to occur immediately, you can use the following process to achieve this.

Set the event property HTML Post for the events that you want to trigger a post. This property is available for the following events:

- Using the Tab key to move out of the text editor: Control Is Exited, Control Is Exited and Changed Inline, Control Is Exited and Changed Asynch
- Using the Tab key to move out of the grid cell editor: Grid Column is Exited, Grid Column Is Exited and Changed Inline, Grid Column is Exited and Changed Asynch
- Selecting a tree node: Tree Node Is Selected, Tree Node Level Is Changed

The solution above is the recommended solution for J.D. Edwards applications. In Xe, if you have already set the event property on desired events, you should also turn off the form property HTML Auto Refresh so that the generator does not need to scan your event rules.

Grid Selection

In a Windows client, the user selects a grid row by clicking the row. The client highlights the selected row in response. In a multiple-selection grid, when the user selects a different row, the previously selected row is automatically deselected unless the user holds down the SHIFT or CTRL key.

In an HTML client, when a grid row is selected it is not highlighted. For single-selection grids, the HTML client shows a radio button beside each grid row. The user clicks the radio buttons to select different grid rows. For multiple-selection grids, the HTML client displays a check box beside each grid row instead.

Note

The HTML client does not automatically deselect the previous grid row if a new row is selected in multiple-selection grids.

Grid Row Attachments

In Windows, by default the grid row does not display the paper clip image, even if it has attachments. When the user hovers the cursor over the row header, the run-time batch program searches to see if an attachment exists for that row. If an attachment exists, the engine displays the paper clip. In an HTML client, nothing is displayed for all grid rows initially, regardless of whether the row has an attachment. When the user clicks the paper clip image in the row header, the run-time batch program searches to see if an attachment exists for that row. If an attachment exists for a given row, the system displays the paper clip image.

Tab Sequence

The HTML client supports tab sequences, with slight differences from the Windows client. Because the HTML client is hosted within a browser, you cannot negate the browser's tabbing sequence. The browser starts tabbing in the page, but after you reach the end of the tab sequence it does not wrap back to the first control in the J.D. Edwards form. Instead, the browser makes a stop in its own URL Address box outside of the J.D. Edwards form.

Grid Functions

The Windows client supports Zoom, Maximize/Restore, Charting, and Print on the grid. These functions are not available in the HTML client.

Type-Ahead Edit

The HTML client uses the type-ahead feature provided by your browser.

Client-Only Business Functions

Client-only business functions are not supported on the Web. All business functions are mapped to the server. A client/server business function or server-only business function should not call client-only business functions.

There is one exception. If an NER is marked client-only solely because it has a form interconnect (for example, it does not access the local hard disk or make Windows-only calls), the NER will function properly on the Web. You should keep these NERs as client-only. When these NERs are generated, J.D. Edwards generates Java classes instead of C code for them. In this context, these types of NERs are called Interpretive NERs.

In summary, J.D. Edwards automatically uses Interpretive NERs when a client-only NER is executed on a server. Consider the following guidelines for Interpretive NERs:

- Interpretive NER can only be called from application ER or from another Interpretive NER.
- Interpretive NER should not be called from UBEs, client/server NERs, or server-only NERs.
- The NER must be designated client-only if it includes form interconnects. J.D. Edwards uses the client-only designation to determine if Interpretive NER should be used.

Multi-Line Edit and Silent Post

By default, the HTML client displays only one editable grid line. You can display multiple edit lines for specific applications. This feature is recommended for high-volume data-entry applications. The multi-line edit feature is a form level feature. You can change it through Form Design.

When a form is set for multi-line edit, the J.D. Edwards run-time batch program also provides a silent post feature. As the user enters data in the grid, the first two lines that the user enters are posted automatically to the JAS server to be processed. The server processes events such as *Grid Row is Exited and Changed-Inline* and *Grid Row is Exited and Changed-Asynch*. Data calculated by the Web server is returned to the Web browser automatically. For example, the associated descriptions are updated to the first two grid lines. At the same time, the user can continue to enter data lines. This feature greatly improves the performance of data entry applications.

Note

The Row is Exited And Changed-Inline event is not processed before the user enters the next row.

Platform Compatibility

When you design Web applications, you should be aware of Web browser compatibility issues. Every browser interprets HTML tags a little differently. Table, form, link, and alignment tags work differently in each browser. Observe the following guidelines when designing your Web applications:

- Design for the appropriate Web browser version. J.D. Edwards Web applications currently run on Internet Explorer 5.5 and subsequent releases and Netscape 6.2 and subsequent releases.
- Avoid using tags supported by only one browser.
- Test designs on multiple browsers and platforms before release.

You should select the screen resolution appropriate to the needs of your target users. For OneWorld Xe, the default form guide size in Form Design Aid is 640 x 480 pixels. Most users with 15-inch monitors are using 800 x 600 screen resolution, and users with 17-inch monitors are using 1024 x 768 screen resolution. B2B users in the manufacturing and distribution industries often use 15-inch monitors with 800 x 600 screen resolution in their facilities.

If you are designing applications for Windows CE or pocket PC, you should select the 240 x 320 pixel option in the Forms Guide menu.

Designing J.D. Edwards Applications for Web Use

When using the J.D. Edwards Form Design Aid (FDA) tool to develop interactive applications targeted for the Web (HTML), you must consider several design strategies during the design and coding process. This topic discusses several of those strategies.

HTML Web applications often appear and function differently from Windows applications. The design requirements dictate how different they are and what the differences are. For example, in Employee, Customer, and Supplier Self-Service applications, one requirement was that non-J.D. Edwards users must be able to navigate between applications and forms. This requirement culminated in the removal of all menu and toolbar exits, which were considered a J.D. Edwards-specific navigation mechanism. To replace menu and toolbar exits, the design allows users to navigate between applications and forms using hyperlinks, which are commonly found in HTML-based applications.

When designing an application for use on both the Windows platform and the HTML platform, you might want to use form modes if you want them to look different. Refer to the *Development Tools Guide* for a detailed explanation of form modes.

When creating menus in J.D. Edwards software to access your new applications, you can access the form in a specific form mode. If your application is Web only and if you did not use form modes in FDA, you can generate the application (HTML serialized objects) in any of these modes. However, if your Web-only application provides form interconnects to other applications that need to be processed in a specific mode, then you must design and generate your new application in that required mode. This is required because of the inheritance of execution modes when a form interconnect is processed. In other words, the mode of the child form is forced to be identical to the mode of the calling form.

Note

All forms accessed from a menu entry point must be generated in the same mode.

Just as when you are creating a Windows application, choosing the appropriate type of form to use is also an essential design consideration. Using the wrong form type causes inefficiencies during run time and user frustration. For example, if you use a form type that supports table updates (for example, headerless detail or header detail) for display-only purposes, the form will refresh (that is, a round trip to the Web server occurs) when the user clicks each grid row.

Performance is also an essential consideration in designing applications for Web use. In addition to the types of performance issues that a Windows environment presents, a Web environment presents additional performance issues to be considered before you perform any coding. When designing applications for Web use, you need to:

- Reduce unnecessary round trips as much as you can by not using event rules with logic that requires immediate processing.
- Separate business logic from user interface logic. For example, you should use only event rules directly from your application to control the user interface on the form. However, you should create business logic components (business functions) that encapsulate business processes such as calculations, database access, and other types of business logic, and that access these components from the application. Doing so makes your application easier to maintain. Reducing the logic in the event

rules also means that your application will be less processor-dependent, thus improving the performance of your application.

- Consider database access performance. *Always set up correct indexes for your tables. Use indexes as much as possible for database access.* Good design practices for three-tier architecture systems advocate separating the logic for database access into individual components (business functions) that are accessed from your application.
- Beware of designs that suppress the display of grid lines in the Grid Record is Fetched or the Write Grid Line – Before events. *If the form is suppressing most or almost all of the grid records that it fetches based on the search criterion, then you should devise a more focused search criterion to avoid having to suppress most grid records.*
- Do not turn off "Page At a Time" processing. If you need the total number of records fetched, you can achieve this by making a JDB call from business functions. "Page at a time" processing offers great performance benefits.
- Do not worry about the round trip from the Web server to the enterprise server. The link between the Web server and enterprise server is usually fast. Additionally, both servers perform smart caching.

Using Form Design Aid to Design Web Applications

Here are some tips and techniques for using the J.D. Edwards Form Design Aid (FDA) to create Web applications.

See Also

- *Generating J.D. Edwards Serialized Objects* in the *J.D. Edwards Java Server Installation Guide* for your platform for information about generating your forms after you have designed them

Designing Forms Using Multiple Modes

You can use control modes to develop an application with multiple interfaces, which reduces the need to maintain several different versions of the same application. You can create one base application and use modes to modify the application for different interfaces. You can enable or hide controls on forms for each mode. Only visibility and enable/disable properties for controls, columns, and menu exits are different for different modes. If you show hidden fields, they appear only for the current mode. All other properties are the same and are common for all modes. All fields are enabled and appear all forms.

The Windows runtime engine does not recognize control modes; only the HTML runtime engine recognizes them. Mode 1 is the default mode. You attach an application to a menu to run. This menu allows you to run an application in different modes. When you run an application over the Web, the application runs in mode 1 by default and another mode if you specify one. If you attach an application to a Windows menu, the Windows runtime engine ignores any modes that you specified and runs the application in mode 1. Use modes consistently throughout your applications. To create Web-enabled versions of your forms, you generate them in Java and HTML using eGenerator. The generator allows you to generate forms simultaneously for one or more modes.

Hiding Menu and Toolbar Exits

If you want to hide form and row exits and all other types of menu and toolbar exits (select, cancel, OK, or delete) in your application, you can do so by selecting the hidden option while you are in mode 2 or 3. If you choose the hidden option in mode 2, then the exit is hidden only in mode 2. In modes 1 and 3, the exit still appears.

► To hide menu and toolbar exits

1. In Form Design and while in mode 2 or 3, on the form with which you are working choose Menu/Toolbar Exits from the Form menu.
2. Choose the exit that you want to hide and click Select.
3. In the State group box, click Hidden, and then click OK.

   When generated in mode 2 or 3 (whichever you were in), the exit will not appear on the form.

Enabling In-Your-Face-Errors

The In-Your-Face-Errors property is a form-level property that is available only for the HTML platform. Typically, the system indicates an application error by highlighting the Errors and Warnings hyperlink in the upper right-hand area of the application. When enabled, however, the In-Your-Face-Errors property causes application errors to appear on the Web page.

► To enable In-Your-Face-Errors

1. In Form Design, do one of the following tasks:
   - Create a new form.
   - Choose an existing form and choose Form Properties from the Form menu.
2. On Form Properties, click Enable In-Your-Face-Errors Display, and then click OK.

Sizing Forms for Screen Sizes

Sometimes you want to size the form according to your target user's browser size. For example, most users of Customer Self-Service applications run in 640 x 480 screen resolutions only.

To create guides for sizing appropriately, you can choose to set the Form Guide value on the Form Properties dialog. Doing so adjusts the blue-line guide in FDA appropriately. FDA does not enforce the size, but merely guides the designer to keep objects within the blue-line guides.

► To size forms for screen sizes

1. In Form Design, do one of the following tasks:
   - Create a new form.
   - Choose an existing form and choose Form Properties from the Form menu.
2. On Find/Browse Form Properties, choose the screen size that you want to use from the Form Guide drop-down menu, and then click OK.

Hiding the Grid Row Selector

The Hide Grid Row Selector property is a grid-level property that is available for the HTML platform. If you decide not to show grid row selectors on your grids, select the Hide Row Selector option on the grid properties dialog box.

► To hide the grid row selector

1. In Form Design, on the form with which you are working, click in the grid and then choose Item Properties from the Edit menu.
2. On Grid Properties, in the HTML Properties section, click Hide Row Selector, and then click OK.

Showing an Alternate Grid Row Format

The Use Alternate Grid Row Format property is a grid-level property that is available for the HTML platform. The property allows you to control the appearance of your grid.

To use this option, you must provide specific HTML tags by clicking the Row Format button. Follow the instructions in the HTML Alternative Grid Row Format dialog when creating your HTML tags for the grid rows. These HTML tags control the appearance of each grid row. You can also choose to output a Media Object Image or Text type object for the grid record and hyperlinks.

This option is supported only in non-updateable grids.

► To show an alternate grid row format

1. In Form Design, on the form with which you are working, click in the grid and then choose Item Properties from the Edit menu.
2. On Grid Properties, in the HTML Properties section, click Use Alternate Grid Row Format, and then click OK.
3. On HTML Alternative Grid Row Format, enter a character string with the HTML tags required to control the appearance of each grid row as required.
4. Click OK.
5. On Grid Properties, click OK.

Using Multi-Line Edit to Control Page Refresh

The Allow Multi-line Edit property is a grid-level property that is available for the HTML platform. This property is applicable only to grids that allow users to edit the grid records. Turning on this option causes the run-time batch program for HTML to show grid rows that contain Editable Text Box controls in each cell (all rows) instead of only in the active grid row. It also prevents the system from refreshing the page every time the user exits a grid row. Instead, the system delays the processing of the Grid Row is Exited events and only processes them in groups of 3 to 5 rows using a silent post.

► **To use multi-line edit to control page refresh**

1. In Form Design, on the form with which you are working, click in the grid and then choose Item Properties from the Edit menu.
2. On Grid Properties, in the HTML Properties section, click Allow Multi-Line Edit, and then click OK.

Controlling the Number of Grid Rows for Each Page of Grid Records

The Grid Row Count property is a grid-level property that is available for the HTML platform. A non-zero value for this option causes the run-time batch program for HTML to show the number of grid rows specified. By default (a value of 0), the run-time batch program displays 10 records in the grid per page.

► **To control the number of grid rows for each page of grid records**

1. In Form Design, on the form with which you are working, click in the grid and then choose Item Properties from the Edit menu.
2. On Grid Properties, in the HTML Properties section, enter the number of grid rows that you want to appear per page in the Grid Row Count field, and then click OK.

Showing Check Boxes in Grid Cells

The Check Box property is a grid column-level property that is available for the HTML platform. It displays check boxes in grid cells for a specific column that provides on/off state information.

To use this option, you must provide specific values that specify the Checked and Un-Checked state value. During run time, the system detects the value of the GC variable for the column and cross-references the value with the settings that you specified to render a checked or unchecked check box.

► **To show check boxes in grid cells**

1. In Form Design, on the form with which you are working, click in the grid and then choose Item Properties from the Edit menu.
2. On Grid Properties, in the Grid Columns section, choose a grid column, and then click Grid Column Properties.
3. On Grid Column Properties, in the Display Style section, click Check Box.
4. Click Values.
5. On Grid Column CheckBox Values, enter numerical values in the Checked and Un-Checked fields, and then click OK.

   The values that you enter should correspond to the values that you set in the GC variable.

6. On Grid Column Properties, click OK.
7. On Grid Properties, click OK.

Showing Hyperlinks in Grid Cells

The Clickable property is a grid column-level property that is available for the HTML and Windows platforms. It allows you to display text in grid cells for a specific column as hyperlinks (clickable text). Any non-blank value in the grid cell for that column appears as a hyperlink.

Enter the logic to process when the hyperlink is clicked in the Grid Column Clicked event. GC values are available for the grid row that was clicked.

► To show hyperlinks in grid cells

1. In Form Design, on the form with which you are working, click in the grid and then choose Item Properties from the Edit menu.
2. On Grid Properties, in the Grid Columns section, choose a grid column and then click Grid Column Properties.
3. On Grid Column Properties, in the Attributes section, click Clickable and then click OK.

   Use the Grid Column Clicked event to define what should occur when the user clicks a hyperlink in this grid column.

4. On Grid Properties, click OK.

Showing Hyperlinks in the Form, Group Box, or Tab Control

To create hyperlinks that are placed in the form, group box, or tab control, you can use either the text block control or static text control. Usually, the text block control is used whenever more control over the text format for the hyperlink is needed; otherwise, the static text control is used due to reduced overhead.

To use a text block control to display its text contents as hyperlinks, you add a segment to hold the text of your hyperlink, and then select the Clickable option in the text block control properties dialog. The logic to process when the hyperlink is clicked should be entered in the Text Clicked event.

To use a static text control to display its text contents as hyperlinks, you assign the hyperlink text that you want to show, and then select the Clickable option in the Static Text properties dialog. The logic to process when the hyperlink is clicked should be entered in the Text Clicked event. You cannot override the font and color for clickable text segments.

► To use the text block control to show a hyperlink

1. In Form Design, on the form with which you are working, select the text box control that you want to affect, and then choose Item Properties from the Edit menu.
2. On Text Block Control Properties, click Add Segment, and enter the text that you want to use as your hyperlink.

   Use the Text Clicked event to define what should occur when the user clicks this hyperlink.

3. In the Segment Information section, click Clickable, and then click OK.

► To use the static text control to show a hyperlink

1. In Form Design, on the form with which you are working, select the static text control that you want to affect, and then choose Item Properties from the Edit menu.
2. On Static Text Properties, enter the text in the Static Text field.

   Use the Text Clicked event to define what should occur when the user clicks this hyperlink.

3. In the Attributes section, click Clickable, and then click OK.

Inserting Custom HTML Tags into a Form

You might want to insert your own HTML into the form to produce a more customized HTML appearance. To do this, use Text Block Control.

The text block control can be used to extend the functionality already provided by the J.D. Edwards HTML platform. The HTML platform adds any text contained in the platform to the HTML of the form as it is generated. In addition, you can add, delete, and update the text within the control at run time. Taken together, this control can greatly extend the functionality of the HTML client.

You can enter text in the text block control either by defining text segments in the text block control in FDA or by inserting text segments at run time via system functions.

Inserting Custom HTML with FDA

Place the Text Control Block on the form, and insert segments as needed to hold the tags. You should segregate segments that are data-driven and segments that contain static HTML tags. Segments that are data-driven can be set during the Dialog is Initialized, Grid Record is Fetched, or other appropriate events by using the Update Segment() system function call.

► To insert custom HTML with FDA

1. In Form Design, choose Text Control Block from the Insert menu.
2. Click in the form in which to place the control.
3. Click the new control, and then choose Item Properties from the Edit menu.
4. Click Add Segment and then enter the HTML text that you want to add to the form.
5. To make the segment clickable by the user, click Clickable.

   This option causes the system to generate the segment as a hyperlink that runs a Text Clicked event when the user clicks it.

6. To override the cascading style sheet (CSS) setting for the segment, perform the following steps:

   a. Turn off Use Default Font and Color.

   b. Click Font and Color.

   c. On Font, specify the typeface, font characteristics, and color that you want to use, and then click OK.

   J.D. Edwards recommends that you use this option sparingly.

7. Click OK.

Inserting Custom HTML at Run Time

You can use the following system functions to manipulate the text block control from ER:

Function	Parameters	Comments
Add Segment	Text Control Text Font Clickable (true/false) SegmentId – Returned unique segment ID for the added segment	
Get Last Clicked Segment	Text Control SegmentId – Returned segment ID of the segment that was last clicked by the user	Use this on the Text Clicked event of the text control to determine what segment was clicked.
Get Segment Information	Text Control Text – Returned text of the passed in segment ID Clickable – Returned clickable flag of the passed in segment ID SegmentId – Segment ID of the segment being inquired about	
Remove Segment	Text Control SegmentId – Segment ID of the segment to be deleted	
Update Segment	Text Control Text Font Clickable SegmentId – Segment ID of the segment to be updated	

The parameters Clickable and SegmentId expect variables of Integer data type.

Advanced Functionality

Except for applying font and color (if specified) and adding the necessary tags for processing the clickable event, the text entered into this control is added unfiltered to the HTML of the form as it is generated. Therefore, you must be familiar with the CSS (Cascading Style Sheet) scheme that J.D. Edwards uses to generate its HTML. By using the appropriate class names in the HTML tags in the text block control, you ensure that the text block control will have the same appearance as the form, and the appearance will change with the rest of the form if the customer changes the controlling CSS forms. This also means that the developer should never define the font and color of text segments unless necessary. The font and color for a text segment ignores the CSS definition of the form and, thus, will always look different from the rest of the form.

Note

The system will not format the text block control correctly in FDA because FDA currently does not communicate with the Web server to determine the correct CSS settings. To see the control correctly formatted, generate the form and then view it on the HTML platform.

The following table lists the common J.D. Edwards CSS tags:

Class Name	Apply to…	Comment
Padded	TABLE	Creates a padded table the width of the page.
Border	Generic	
NoBorder	Generic	
WideTable	Generic	Width = 100%
TallTable	Generic	Height = 100%
TallAndWideTable	Generic	Width = Height = 100%
Grid	TABLE	
MainHeading	Generic	
SectionHeading	Generic	
GroupHeading	Generic	
SubHeading	Generic	
FieldLabel	Generic	
RaisedBorders	Generic	
BlackBorders	Generic	
ClearBorders	Generic	
QBECell	Generic	
GridHeaderCell	Generic	

Class Name	Apply to…	Comment
GridCell	Generic	
InYourFaceError	Generic	
InYourFaceWarning	Generic	
ToolbarText	Generic	
GroupBox	Generic	
GroupBoxHeader	Generic	
FormLabel	Generic	
FormAboveGrid	Generic	

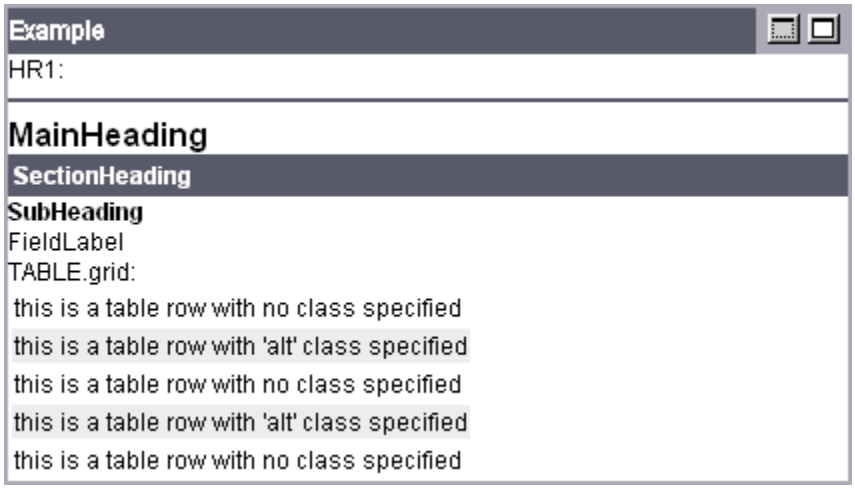
The following HTML tags have been specified with custom style tags so that they can be used with the assurance that their text will be formatted correctly:

Tag	Comment
HR1	
HR2	
BODY	Non-formatted text without any enclosing tags, classes, or both entered by the text block control will have a base style to rely on.
INPUT	
SELECT	
A	
TABLE	
TABLE TD	

The following sample HTML code produces the figure that immediately follows it:

```
Text in text block control :
<HR1>
HR1:
</HR1>
<HR class=MainHeading>
MainHeading
</HR>
<HR class=SectionHeading>
SectionHeading
</HR>
<HR class=SubHeading>
```

```
SubHeading
</HR>
<span class=FieldLabel>
SubHeading
TABLE.grid:
</span>
<TABLE class=Grid>
<TR><TD >
this is a table row with no class specified
</TD></TR>
<TR class=alt><TD >
this is a table row with no class specified
</TD></TR>
<TR><TD >
this is a table row with no class specified
</TD></TR>
<TR class=alt><TD >
this is a table row with no class specified
</TD></TR>
<TR><TD >
this is a table row with no class specified
</TD></TR>
```



Inserting Images into a Form

You can place a variety of images on a form with the Bitmap Control. The image can be static, or you can make it clickable. Use the Button Clicked event to define what should occur when the user clicks the image.

► To insert images into a form

1. In Form Design, choose Bitmap from the Insert menu.
2. Click in the form in which to place the control.
3. On Bitmap Properties, click Find to find the file that you want to use for your bitmap control.

   The Bitmap Properties form displays the bitmap that you have chosen.

4. Complete the following field:
   - Tool Tip Text

     This text, like Hover Helps, appears if the user makes the mouse pointer hover over the bitmap.

5. Click one or more of the following attributes:
   - Clickable

     If you turn on the Clickable option, a Button Clicked event is enabled for the bitmap.

   - Maintain Aspect Ratio

     If you turn on the Maintain Aspect Ratio option, the bitmap will maintain its dimensions if you resize it. This ensures that your image does not become distorted if you resize it.

Inserting Drop-Down Lists into a Form

To display drop-down lists in a form, you should place the combo box control on the form. Then, drag and drop a data dictionary item right on the combo box control. When correctly done, your Combo Box Properties dialog will state the data dictionary item to which it is connected.

The content of the drop-down list is derived from (and only from) the UDC values and descriptions with which the data dictionary item is associated. The list is based on the edit rule properties of the data dictionary item. Allowing only a UDC to complete a combo box ensures that all the combo box values can be translated correctly.

► To insert drop-down lists into a form

1. In Form Design, choose Combo Box from the Insert menu.
2. Click in the form in which to place the control.
3. From the View menu, choose Data Dictionary Browser.
4. In Data Dictionary Browser, find the data dictionary item that you want to use for the combo box control.

5. Select the combo box control, and then drag the data dictionary item from the Data Dictionary Browser and drop it directly on top of the combo box control.
6. Select the combo box control, and then choose Item Properties from the Edit menu.
7. Verify that the data dictionary item has been associated with the combo box and click OK.

Performing Custom Selects and Sorts for the Grid

Within the Button Clicked event of the Find menu exit, you can customize your grid selection criterion by using the Clear Select() system function and then performing the Set Selection() system function. You can also choose to perform Clear Sequencing() and then the Set Sequencing() system function to customize the grid sequencing settings of the grid.

By using Set Selection() or Set Sequencing() system functions, you do not have to use hidden filter fields on the form to perform custom grid-selection criteria. Furthermore, the Set Selection() function can support “Or” and “And” statements in the resulting SQL statements.

Finally, the Set Selection() system function does not depend on fields that exist in the business view of the form. Consequently, you can actually perform selects on fields that are outside of the business view, so long as they belong to the same table as the business view.

J.D. Edwards Software Events on a Web Client

Two J.D. Edwards software events do not work well in an HTML environment because of posting. J.D. Edwards recommends that you do not use the Control Is Exited event to hide or show controls or to enable or disable grids. If you do use Hide/Show or Enable/Disable Grid system functions in the Control Is Exited event, the system automatically generates JavaScript to refresh the page so that the event can be processed immediately. Note that this solution causes a round trip.

The following table shows how an HTML client responds to all J.D. Edwards software events, including which ones trigger a post:

Object Type	Event	Triggers Post	Delayed
Bitmap	Button Clicked	Yes	No
Checkbox	Selection Changed	No, but will trigger post if ER is present.	—
Combo Box	Control is Exited	No	Yes
ComboBox	Control is Entered	No	Yes; will run against state of form prior to post.
Document Object	Media Objects — Form def	— (Triggered on server during form initialization.)	—
Document Object	Media Objects — Row def	— (Triggered on server during Find button processing.)	—

Object Type	Event	Triggers Post	Delayed
Edit	Control Exited/Changed—Asynch	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes
Edit	Control Exited/Changed—Inline	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes
Edit	Control is Entered	No	Yes; will run against state of form prior to post.
Edit	Control is Exited	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes
Edit	Post Visual Assist Clicked	— (Triggered on server immediately after the event, Visual Assist Button Clicked, is finished.)	—
Edit	Visual Assist Button Clicked	Yes	No
Form	Add Record to DB — After	— (Triggered on server during OK button processing.)	—
Form	Add Record to DB – Before	— (Triggered on server during OK button processing.)	—
Form	Clear Screen After Add	— (Triggered on server during form initialization.)	—
Form	Clear Screen Before Add	— (Triggered on server during form initialization.)	—
Form	Dialog is Initialized	— (Triggered on server during form initialization.)	—
Form	Director Back Button Clicked	N/A (Not valid on the HTML client.)	N/A
Form	Director Cancel Button Clicked	N/A (Not valid on the HTML client.)	N/A
Form	Director Next Button Clicked	N/A (Not valid on the HTML client.)	N/A
Form	End Dialog	— (Triggered on server during form closing.)	—
Form	Grid Record is Fetched	— (Triggered on server during Find button processing.)	—
Form	Last Grid Record Has Been Read	— (Triggered on server during Find button processing.)	—
Form	Post Commit	— (Triggered on server during Find button processing.)	—

Object Type	Event	Triggers Post	Delayed
Form	Post Dialog is Initialized	— (Triggered on server during form initialization.)	—
Form	Update Record to DB — After	— (Triggered on server during OK button processing.)	—
Form	Update Record to DB — Before	— (Triggered on server during OK button processing.)	—
Form	Write Grid Line—After	— (Triggered on server during OK button processing.)	—
Form	Write Grid Line—Before	— (Triggered on server during OK button processing.)	—
Grid	Add Grid Rec to DB — After	— (Triggered on server during OK button processing.)	—
Grid	Add Grid Rec to DB — Before	— (Triggered on server during OK button processing.)	—
Grid	Add Last Entry Row to Grid	— (Triggered on server during Find button processing.)	—
Grid	All Grid Recs Added to DB	— (Triggered on server during OK button processing.)	—
Grid	All Grid Recs Deleted from DB	— (Triggered on server during OK button processing.)	—
Grid	All Grid Recs Updated to DB	— (Triggered on server during OK button processing.)	—
Grid	Delete Grid Rec from DB—After	— (Triggered on server during OK button processing.)	—
Grid	Delete Grid Rec from DB—Before	— (Triggered on server during OK button processing.)	—
Grid	Delete Grid Rec Verify—After	— (Triggered on server during Delete button processing.)	—
Grid	Delete Grid Rec Verify—Before	— (Triggered on server during Delete button processing.)	—
Grid	Double Click on Row Header	Yes	No
Grid	Get Custom Grid Row	— (Triggered on server during Find button processing.)	—
Grid	Kill Focus on Grid	No	Yes

Object Type	Event	Triggers Post	Delayed
Grid	Row Exit & Changed — Asynch	Yes. If the grid has the flag, Allow Multi-line Edit, turned on, the post occurs in the background without refreshing the form, making it appear that the post did not happen when it actually did.	No
Grid	Row Exit & Changed – Inline	Yes. If the grid has the flag, Allow Multi-line Edit, turned on, the post occurs in the background without refreshing the form, making it appear that the post did not happen when it actually did.	No
Grid	Row is Entered	No	Yes; will run against state of form prior to post.
Grid	Row is Exited	Yes. If the grid has the flag, Allow Multi-line Edit, turned on, the post occurs in the background without refreshing the form, making it appear that the post did not happen when it actually did.	No
Grid	Set Focus on Grid	Sometimes. A post occurs if the user clicks a cell in an editable grid. The event is delayed if the user tabs into the editable grid.	Sometimes. A post occurs if the user clicks a cell in an editable grid. The event is delayed if the user tabs into the editable grid.
Grid	Update Grid Rec to DB—After	— (Triggered on server during OK button processing.)	—
Grid	Update Grid Rec to DB—Before	— (Triggered on server during OK button processing.)	—
Grid Column	Col Exited & Changed – Asynch	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes
Grid Column	Col Exited & Changed — Inline	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes

Object Type	Event	Triggers Post	Delayed
Grid Column	Col is Exited	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes
Grid Column	Grid Column Clicked	Yes	No
Grid Column	Post Visual Assist Clicked	— (Triggered on server immediately after the event, Visual Assist Button Clicked, is finished.)	—
Grid Column	Visual Assist Button Clicked	Yes	No
Hyper Item	Button Clicked	Yes	No
Hyper Item	Post Button Click — Asynch	—— (Triggered on server immediately after the hyper item's event, Button Clicked, occurs.)	—
Hyper Item	Post Button Clicked	— (Triggered on server immediately after the hyper item's event, Button Clicked, occurs.)	—
Parent/Child	Kill Focus On Control	No	Yes
Parent/Child	Set Focus on Control	No	Yes
Parent/Child	Tree — Begin Drag Operation	Yes	No
Parent/Child	Tree — Cancel Drag Drop	Yes	No
Parent/Child	Tree — Drag Over Node	Yes	No
Parent/Child	Tree — End Drag Drop Operation	Yes	No
Parent/Child	Tree Node Is Collapsing	Yes	No
Parent/Child	Tree Node Is Deleted	Yes	No
Parent/Child	Tree Node Is Expanding	Yes	No
Parent/Child	Tree Node Selection Change	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes
Parent/Child	Tree—Node Level Changed	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes
Parent/Child — Grid	All Grid Recs Deleted from DB	— (Triggered on server during OK button processing.)	—
Parent/Child — Grid	Delete Grid Rec from DB—After	— (Triggered on server during OK button processing.)	—
Parent/Child — Grid	Delete Grid Rec from DB—Before	— (Triggered on server during OK button processing.)	—

Object Type	Event	Triggers Post	Delayed
Parent/Child — Grid	Delete Grid Rec Verify—After	— (Triggered on server during Delete button processing.)	—
Parent/Child — Grid	Delete Grid Rec Verify—Before	— (Triggered on server during Delete button processing.)	—
Parent/Child — Grid	Double Click on Row Header	Yes	No
Parent/Child — Grid	Get Custom Tree Node	— (Triggered on server during Find button processing.)	—
Pushbutton	Button Clicked	Yes	No
Pushbutton	Post Button Clicked	— (Triggered on server immediately after the push button's event, Button Clicked, occurs.)	—
Radio button	Selection Changed	No, but will trigger post if ER is present.	—
Static Control	Text Clicked	Yes	No
Tab Page	Tab Page is Initialized	— (Triggered on server before the event, Tab Page is Selected, is processed, provided that the tab page is selected for the first time.)	—
Tab Page	Tab Page is Selected	Yes	No
Text Block	Text Clicked	Yes	No
Tree Control	Double Click on Leaf Node	No, but will trigger post if ER is present.	—
Tree Control	Get Custom Tree Node	— (Triggered on server during the tree control's event, Tree Node Is Expanding.)	—
Tree Control	Kill Focus On Tree	No	Yes
Tree Control	Set Focus On Tree	No	Yes
Tree Control	Tree Node Is Collapsing	Yes	No
Tree Control	Tree Node Is Deleted	Yes	No
Tree Control	Tree Node Is Expanding	Yes	No
Tree Control	Tree Node Selected	No, but will trigger post if the event flag, HTML Post, is turned on.	Yes

Designing J.D. Edwards Applications for Mobile Use

From SP15 onwards, the J.D. Edwards HTML client provides full support for mobile devices based on the Windows CE platform, which is highly compatible with the HTML client architecture. Currently, all of the Window CE devices have a good support for HTML through various versions of Internet Explorer (IE) browser.

J.D. Edwards Application support for mobile devices includes the following applications:

- Windows CE 3.0 devices running IE 4.x Browser
  These devices include HP Jornada 720, NEC Mobile Pro™ 790, and so forth. Version 3.0 represents the latest version of Windows CE, and the devices based on it are recommended over Windows CE 2.11 because Version 3.0 provides support for the HTML4 specification by the IE 4.x Browser. IE 4.x is the richest browser currently available on mobile devices and includes good support for DTML and JavaScript.
- Windows CE 2.11 SP1 and above running IE 3.01
  J.D. Edwards HTML Client supports these devices. However, these devices are not recommended because they represent the older technology that has been superceded by Windows CE 3.0.
- Windows CE 3.0 running Pocket PC 2000
  These devices include Compaq IPAQ, HP Jornada 540, Casio Cassiopeia, and so forth. These devices also provide HTML access based on the IE3.01 browser. The J.D. Edwards HTML Client fully supports these devices.

The J.D. Edwards Mobile Device architecture supports all J.D. Edwards interactive applications that are run on the supported platforms. It leverages the existing architecture and requires that you make no application development changes to your applications. Currently, J.D. Edwards Mobile support is available for connected users only; that is, users who are not connected to the network via a browser are not supported. Support for the Palm Computing platform is also unavailable because of its lack of good browser support.

Mobile Device Run-time Architecture

The architecture providing support for mobile devices in the J.D. Edwards HTML client is the same as the one for desktop. The output generation for the HTML4-compliant browsers remains essentially the same. However, output is generated for Windows CE browsers running IE3.01.

When the user first signs onto J.D. Edwards JAS server, a session is established between the client browser and the Web server. The session then holds information about the client platform and the browser. The system generates HTML output based on this information.

Mobile Device Design Strategies

When you develop applications for mobile devices, use the same design strategies as for regular Web-based applications. However, remember that the form factors are limiting on these devices. The form factor on Window CE 3.0 devices is limited to 640 x 480, while the form factor on Pocket PC is limited to 240 x 320. The following list presents additional design strategies to consider:

- In browser-based applications, vertical scrolling is generally more acceptable than horizontal scrolling.

- Try stacking controls top-to-bottom for the Pocket PC interface to make the applications more useful. You can limit the number of grid columns to prevent the user from scrolling too much.
- From SP16.1 forward, J.D. Edwards HTML supports grid tabs on HTML4 compliant browsers. However, grid tabs are not supported on Pocket PCs. Grid tabs support a feature called Default for Pervasive Device. You can use this feature to limit the number of grid columns displayed on Pocket PC for existing applications.

► To use a specific grid format for mobile devices

1. On any form in HTML showing an editable grid, click Customize Grid.
2. On Customize Grid, choose a format from the Available Formats list.
3. To make a grid format the default format for mobile devices, click Default for Pervasive Devise.
4. Click Close.

Functional Differences between HTML and Mobile Devices

Mobile devices manage errors differently than HTML. Also, the user interacts with a mobile device differently from the way he or she interacts with a Web browser displayed on a terminal.

Event Handling

The J.D. Edwards HTML Client and Mobile Device manage form events similarly. However, on the Pocket PC, the Control Is Exited event is never processed for a control until the user changes the data in the associated field. In HTML 4-compliant browsers such as IE 5.x on a desktop and IE 4.x on Windows CE 3.0, JavaScript is used to keep track of all the controls that the user has used the Tab key to move out of. This information is posted to the Web server and used to run the Control Is Exited event. However, on the IE 3.01 browser on Pocket PC, the virtual client keeps a virtual image of the Form, and this image is used to compare the fields that are changed. Therefore, Control Is Exited and its associated events (such as Control is Entered, Control is Exited, and Changed) are processed only if a field is modified. The logic of your application should not depend on Control Is Exited to be processed even if a particular field is not modified.

Usability

The appearance of applications running on mobile devices is not as rich as the one on Desktop IE 5.x browser because the DHTML and JavaScript support on the browsers for these devices is limited. The following is a list of the most important differences:

- Grid scrollbar support does not exist on any mobile device.
- The HTML Client on a Pocket PC has limited keyboard support.
- Multi-line Grid Editing functionality does not exist on the Pocket PC. Therefore, these devices are not suitable for high-volume data entry.
- Support for Media Object RTF Editing and OLE Objects does not exist.
- No Export/Import functionality is supported on any mobile device.

- Support for text block controls is limited by the mobile device. If an application has a text block control that relies heavily on DHTML/JavaScript, it might not work correctly on Pocket PC platforms.
- Support for Viewing PDF files for submitted reports does not exist.

Multiple Application Framework

Frequently, end users need to be able to work in several applications simultaneously. If this is true of your Web client users, then you should use the J.D. Edwards Multiple Application Framework (MAF) to launch applications. MAF enables multiple, simultaneous browser sessions, and matches run-time data to the correct application.

MAF acts as the coordinator between the calling source (such as Task Explorer or the J.D. Edwards Portal) and the target application. Through the use of the MAF Launcher, MAF is able to facilitate the transfer of data from the source to the target, despite language mismatches or other operational hurdles. J.D. Edwards provides an MAF Launcher that already matches anchors for Portal and Task Explorer sources to Java objects containing resource bundles for J.D. Edwards ERP application targets.

If you launch an application through MAF, the MAF facilitates the exchange of data between source and target. Additionally, if the user closes the browser containing the source, then the user can later launch a new browser containing the source from the target browser. Finally, if a user terminates a target by closing the browser, MAF removes the corresponding pointer to the source object in the queue that it maintains to link sources to targets.

To take advantage of the system, you must launch your application through MAF.

Note

Applications are referred to as components in the MAF code structure.

Implementing an MAF component

Implementing an MAF component is done by extending two abstract classes: MAFComponent and JASMAFComponentInstance. You can allow your implementation of MAFComponent to contain your implementation of the JASMAFComponentInstance as an inner class because the Component Instance will never be created outside of the context of the MAFComponent. Additionally, you can keep all of your code for a particular component in one place.

Extending the MAFComponent

The following code extends the MAFComponentObject. The example is not aware of language. (The newMAFComponent method is documented later.)

```
import com.jdedwards.base.maf.*;

import com.jdedwards.base.maf.jas.*;

import java.util.*;


public class MyMAFComponent extends MAFComponent

{

public String getId()
```

```
{
    return "JDE_MAF_EXAMPLE";
}
public String getDescription(Locale locale)
{
    //This is not language aware but is just an example
    return "MAF Example Component";
}
…
}
```

The first method is the unique Id for this component type. The second returns a description. Notice that the second method provides a locale. This can be used to provide either ISO compatible language codes or to retrieve information from a resource bundle. If you are looking for a J.D. Edwards Language Preference, you must look up the ISO code and translate it to the J.D. Edwards language code.

Extending JASMAFComponentInstance

The following code is an implementation of the JASMAFComponentInstance class as well as the newComponentInstance method from the example above. The JASMAFComponentInstance implementation is an inner-class of the MAFComponent implementation in this example.

```
import com.jdedwards.base.maf.*;
import com.jdedwards.base.maf.jas.*;
import java.util.*;

public class MyMAFComponent extends MAFComponent
{
…
public MAFComponentInstance newMAFComponentInstance(HttpSession session,
String uniqueId, String launcher)
{
    return new MyMAFComponentInstance(uniqueId, launcher, this);
}

private class MyMAFComponentInstance extends JASMAFComponentInstance
{
public MyMAFComponentInstance(String uniqueId, String launcher, MAFComponent
component)
{
    super (uniqueId, launcher, component);
```

```
}
public String getData(HttpServletRequest req, HttpServletResponse resp)
{
    return "This is my component's data";
}
public String getDescription (Locale locale)
{
    return "My component instance: "+getUniqueId();
}
}
}
```

Registering the MAFComponent

Before an MAF Component can be used, it must be registered with the dynamic MAFComponentFactory class. Because the MAFComponent in question might be needed before the component actually is processed, you should register the component during initialization of the WebApp. The following lines of code will instantiate and register the MAFComponent above:

```
MAFComponent comp = new MyMAFComponent();
MAFComponentFactory.register(comp);
```

Obtaining and Using the MAFComponentInfo

The MAFComponentInfo must be provided to your object from the MAF launcher except during initialization and destruction of the instance. MAFComponentInfo will be on the HttpServletRequest object in the attributes. You obtain this object by using the following code:

```
MAFComponentInfo info =
request.getAttribute(MAFConstants.ATTRIB_COMPONENT_INFO);
```

Generating a MAF Close link

MAF provides a link that is used to close everything in the instance. This includes removing your component from the Open Components Map and processing the MAFComponentInstance's destroy method. To generate this link (which could be a javascript), use the following code:

```
out.println("<a href=\""+info.getCloseAnchor()+"\">Close</a>");
```

If you need to add extra functionality to the close, then you should implement the destroy method.

Reconnecting with a Closed Launcher

Your component can provide a link back to its launcher based on information in the associated MAFComponentInfo object. This ability is optional and your component should search for an active link before generating the tag. The following code establishes a Launcher reconnect link:

```
out.println("<a href=\"" + info.getLauncherAnchor() + "\">Goto " +
info.getLauncherDesc(locale) + "</a>");
```

Obtaining Your Unique ID

The backbone of the MAF system is in the MAF Unique Id. For any given session, this will allow you to determine which instance of your component is actually being processed. You can obtain:

```
String uid = info.getUniqueId();
```

Performing a Loop-Through

MAF needs to monitor the component after it has started to facilitate external windows. You can perform this loop-through in one of two ways. The first and simplest technique is to use the wrapURL method on the MAFComponentInfo object. This technique is flexible enough to handle static links only, however. Use the wrapURL method as shown below:

```
Map params = new HashMap();

params.add("myParam", {"oh me", "oh my"});

out.println("<a href=\"" + info.wrapURL(params) + "\">my url</a>");
```

This code generates a tag similar to the following example:

```
<a
href="http://jde.com/jde/servlet/com.jdedwards.base.maf.jas.JA
SMAFServlet?…&myParam=oh+me&myParam=oh+my">my url</a>
```

Notice that the specified parameters appear at the end of the URL in the anchor. Those parameters are sent back to your component using the getData method in your MAFComponentInstance.

The second technique is to use the getMAFLauncherParameterBean provided by the MAFComponentInfo. This provides the URL and parameters for the component in object form. Although you can construct a static URL with this object, its strength lies in the ability to generate links for form submissions. Consider the following code:

```
MAFLauncherParameterBean mlpb = info.getMAFLauncherParameter(new HashMap());

out.println("<FORM action=\"+launcherParamBean.getURL()+"\">");


Map m = launcherParamBean.getParameterMap();

Iterator iterator = m.keySet().iterator();

while(iterator.hasNext())

{

  String paramName = (String)iterator.next();

  String[] paramValues = (String[])m.get(paramName);

  for (int x = 0; x < paramValues.length; x++)

  {
```

```
     out.println("<INPUT type=\"hidden\" name=\""+paramName+"\"
value=\""+paramValues[x]+"\">");
  }
 }
 out.println("Name: <input type=\"text\" name=\"name\">");
 out.println("</FORM>");
```

This code generates a form with all of the parameters required by the MAF as hidden fields as well as a Name property that the user could type.

Although more work than the wrapURL method, the getMAFLauncherParameterBean provides considerably more flexibility. These parameters will be on the request when your MAFComponentInstances' getData method is accessed. MAF uses a simple parameter pass-through system to forward the proper request on to the component.

Web Client Configuration

When you set up the JAS server, you performed most of the configuration work that you need to do for the Web client. See your JAS documentation for information about settings for .ini files and for performance tuning your system.

Additionally, you have several choices about how you configure system login. If you are also using the J.D. Edwards Portal, any login configuration you do will affect both systems.

Configuring Login

After establishing access to the system and its features with Security Workbench, users with access rights login using their J.D. Edwards ID and password. You can define the language to use for the login forms (assuming translated versions of the forms exist). In the JAS.ini file, in the OWWEB section, set InitialLanguageCode to the ISO code representing the initial language you want to use. This setting overrides any local language preferences until the user is logged.

You can make use of cookies to allow a local workstation to record the details of a user's login (including password). Doing so streamlines the login process for the user. You can also create an anonymous user account to allow users to login anonymously.

If desired, you can configure the system so that instead of using the standard login feature, you can use a simplified basic authentication procedure instead.

In addition to indicating when a user's password has expired, the system can notify users of an impending expiration, depending on the user's login method. A user can elect to change his or her password at the time of notification or can ignore the warning. If the user changes his or her password at the time of notification, then the system logs the user out and back in again with the new password.

The following login methods do not support password condition notification:

- Anonymous
- Parameter-based
- Direct (via cookie)

Notification of impending password expiration occurs automatically by default. If you want to suppress this notification, set the JAS.ini parameter, DisablePasswordAboutToExpire (in the [LOGIN] section), to TRUE.

Anonymous Users

J.D. Edwards software accepts an anonymous user login if the JAS server running the system instance has been configured to accept it and if the local JAS.ini file has been configured properly. You establish access rights to workspaces and components for the anonymous login as a separate user. The anonymous user is not a part of the *PUBLIC group. If you choose to establish an anonymous user, you must explicitly grant it access to system objects.

An instance that has been configured to accept anonymous users behaves differently from a regular instance. When a user browses to it, the system displays the initial Welcome form

instead of requiring a login. The user can view objects to which the anonymous user account has been granted access. The user cannot perform any command actions such as changing user options, although if the user has an account, the user can choose to login normally. After logging in, the user has access to all objects and command functions to which his or her account is entitled.

The anonymous user account is a valid account, and a user can use it to log in to the system in the same way that a user can log in using a regular account. You can grant the anonymous user account access to command functions and other features just as you can a normal account. A user who logs in using the anonymous account can then access the command actions and features you granted the account.

► To configure an instance of the system on the JAS server to accept an anonymous user login

1. Create a specific, unique user account in J.D. Edwards software to be used as the anonymous user account.
2. Add the settings to the command line parameters appropriate to your environment:
   - For the JVM for Tomcat/WebSphere 3.5, add these settings:
     - Danon.user.ouid=*USR12345*

       where *USR12345* is a specific, unique user ID that you set up expressly to be the anonymous user account.

     - Danon.user.pwd=*PASS12345*

       where *PASS12345* is the password that you want to use for the anonymous user account.

     - Danon.user.env=*ENV12345*

       where *ENV12345* is the instance of the system for which you want to allow anonymous logins.

     If you are using Tomcat, add these parameters to Project|Project Properties|Run|VM Parameters.

   - For WebSphere 4.0, add the following runtime variables for the Web server:

Name	Value
anon.user.ouid	A specific, unique user ID that you set up expressly to be the anonymous user account.
anon.user.pwd	The password that you want to use for the anonymous user account.
anon.user.env	The instance of the system for which you want to allow anonymous logins.

3. Update the command line arguments as appropriate to your environment:
   a. If you are using WebSphere 3.5, use the WebSphere Administrative Console and go to the Application Server menu level to update the Command Line Arguments under the General tab as follows:

      ```
      -Danon.user.ouid=[USR] -Danon.user.pwd=[PWD] -
      Danon.user.env=[ENV]
      ```

   b. If you are using WebSphere 4, complete these steps:

i. Navigate to the JVM Settings tab of the Application Server.

ii. Add the `anon.user.ouid`, `anon.user.pwd` and `anon.user.env` entries with their respective values in the System Properties section.

iii. Click Add, and then click Apply.

The system updates the command line arguments.

4. Add the following line to the [OWWEB] section of the JAS.ini file:

```
AnonAccess=true
```

5. To create a hyperlink that allows the user to launch a regular Portal login process, add the following line to the [PORTALCONFIGURATION] section of the JAS.ini file:

```
ShowSignin=TRUE
```

When you enable this feature, the login hyperlink appears on the Workspace Navigation Bar. A user logging in anonymously can use it to launch the regular Portal login process. In this way, a user with a Portal account who logs in anonymously can then log in as a regular user.

6. Restart the application server instance on the server for the changes to take effect.

Direct Login

You can use cookies to record a user's user name, environment, role, language, and even password to streamline the login process the next time the user logs in.

To configure the system to record user name, environment, role, and language (but not password), add or edit the following line in the JAS.ini file in the SECURITY section:

```
UseLogonCookie=TRUE
```

If the user logs on before the cookie expires, he or she is prompted only for a password.

To configure the system to record all user-related information, including password, add or edit the following line in the JAS.ini file in the SECURITY section:

```
UseLogonCookie=DIRECT
```

If the user logs on before the cookie expires, the login process is transparent to the user, and the system appears to launch directly. With this setting, a standard encryption key is used when the system records the password. However, you can use your own encryption key when writing passwords. To set an encryption key, add or edit the following line in the JAS.ini file in the LOGIN section:

```
PassKey=[alphanumeric key]
```

You can also set cookie expiration time. When the cookie expires, its information is deleted. If a user logs in after his or cookie has expired, the user must provide login information again. To set cookie expiration time, add or edit the following line in the JAS.ini file in the SECURITY section:

```
CookieLifeTime=[time in days]
```

Basic Authentication

You can configure the system to use a generic web login process instead of the default login process. Although basic authentication streamlines the login process, the user cannot specify role or environment. Consequently, when enabled, all users log in to the default environment with the default role.

Typically, basic authentication is used by third-party products to log into the Portal. For a third-party product to use basic authentication, the product must submit a basic authentication header to the Portal.

To enable basic authentication, add or edit the following lines to the specified sections of the server's JAS.ini file:

Section	Setting
SECURITY	SSOEnabled=TRUE
OWWEB	DefaultEnvironment=[an environment]
OWWEB	DefaultRole=[a role]