Plug-In User Interface Guidelines

vSphere Client

This document provides guidelines for vSphere Client user interface look and feel when developing plug-ins. Adhering to these guidelines ensures that your vSphere Client plug-in will look like other plug-ins provided by VMware and other vendors that adhere to these guidelines.

This document covers the following topics:

General Conventions

The following conventions apply to all areas of the user interface.

Text

These conventions apply to text:

Element	Convention
Text	Tahoma regular, title capitalization is used for form fields and labels, 8pt 11px 0.7em 70%
Dialog Text	Configured Windows font for dialogs (by default is MS San Serif for Win2k, Tahoma for Win XP). Size is 8pt.
Link Color	#0066FF
Form Field Labels and Headings Case	Capitalize all words

Widgets

vSphere Client widgets consist of buttons, check boxes, and combo boxes.

Buttons

You can use system style or adhere to the following conventions.

Element	Convention
Text	Tahoma regular, title capitalization, 8pt 11px 0.7em 70%, Black (#000000)
Background Color	Gray (#CCCCCC)
Border Color	Gray Bevel (#aab3b3)
Corner Radius	1, 90 degrees

Check Boxes

Use system style.

Combo Boxes

Use system style.

Views

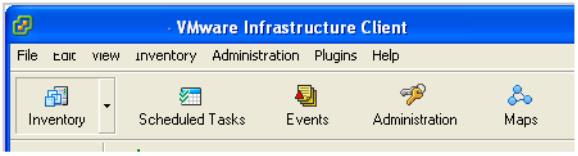
The following conventions apply to views.

Launching a View

The following conventions apply to the layout of views.

Views Toolbar

The Views toolbar is relevant to releases up to VI Client 2.5. The Views toolbar serves as a launching point for all applications in the VI Client.



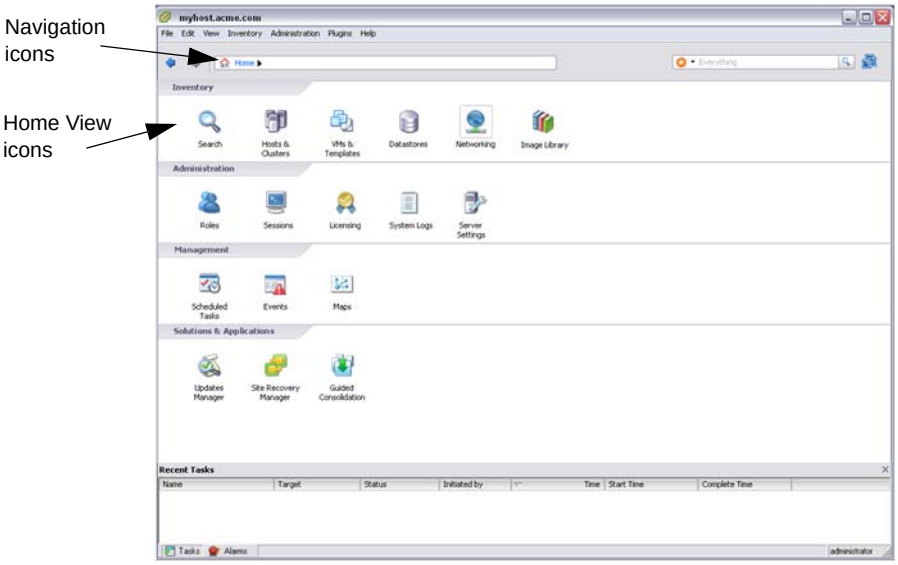
Applications that encompass a full view must provide an icon and application name.

These conventions apply to the Views toolbar:

Element	Convention
Toolbar Icon Size	20 X 20px canvas. 18 X 18px graphic.
Plug-in Name String	Up to 28 characters

Home View

For VMware vSphere Client 4.0, the Home View replaces the Views toolbar. The Home View serves as a launching point for all applications in the vSphere Client.



Applications that encompass a full view must provide an icon and application name. The **Solutions & Applications** section includes plug-in applications.

These conventions apply to the Home View:

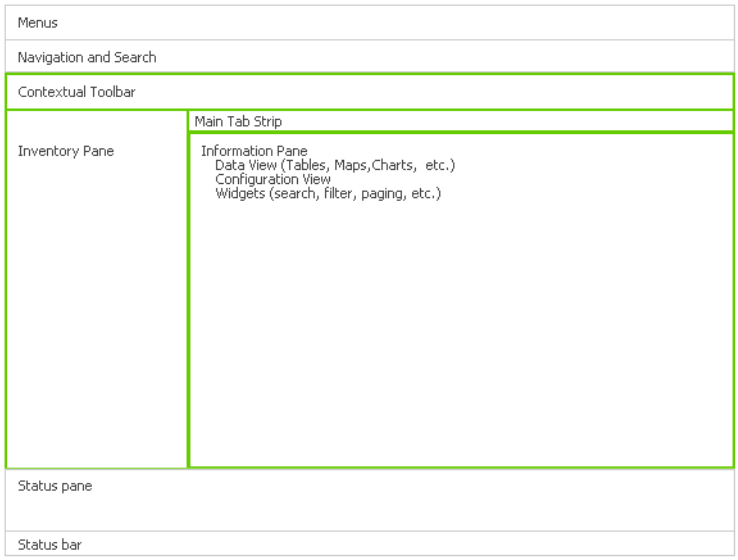
Element	Convention
Home View Icon Size	20 X 20px
Navigation Icons Size	16 X 16px
Plug-in Name String	Up to 28 characters

View Layout and Interaction

The plug-in creator must determine what components best suit the users' needs. Most views in the vSphere Client adhere to the following anatomy.

- Menu
- Navigation and Search
- Contextual toolbar
- Inventory pane
- Main tab strip
- Information pane

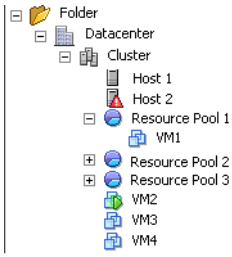
For plug-ins, the green areas in the figure below are specified by the plug-in creator.



Contextual Toolbar

Inventory Pane

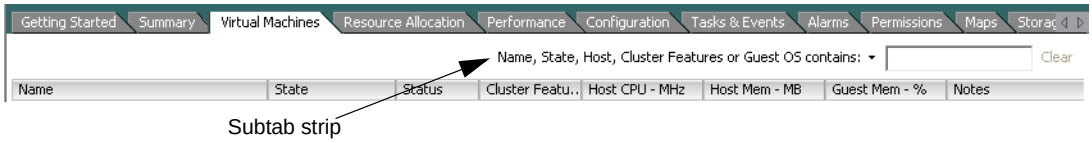
The inventory pane is optional. The inventory pane displays a hierarchical list of monitored VMware Infrastructure objects. Monitored objects include datacenters, resource pools, clusters, networks, data stores, templates, hosts, and virtual machines.



Main Tab Strip and Secondary Strip

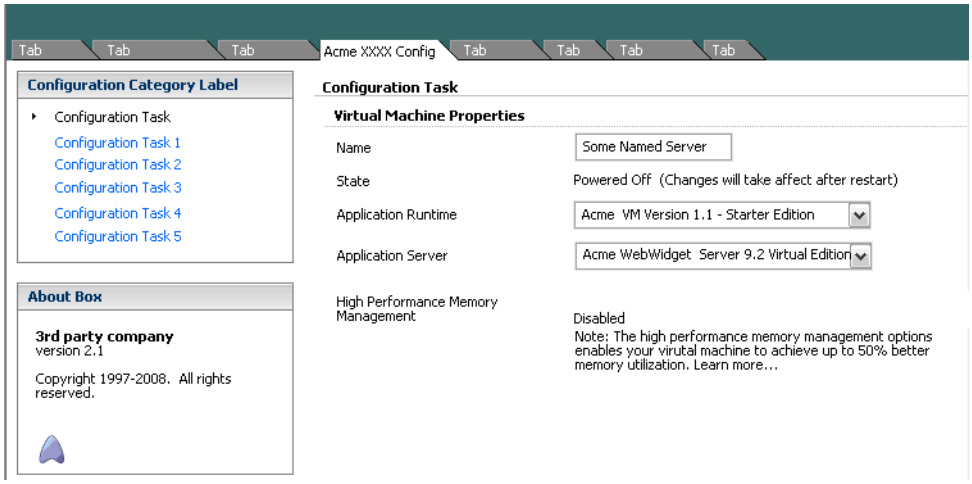
The main tab strip allows you to create different views of the same information or object. For more information, see "Main Tab Strip" on page 5.

The subtab strip houses command links, filters, time/date stamps, sub-tabs, and other widgets that apply to a specific view. For more information, see "Subtab Strip" on page 6.



Information Pane

The information pane is required. Information panes display configurations, lists and charts relevant to a particular task or inventory object. The information pane may be divided into two or more tabbed elements defined by the main tab strip.



Tab Strips

The following conventions relate to tab strips.

Main Tab Strip

You can use the tabs to create different views of the same information or object. For example, you can display the summary, configuration, details, and map relationships for virtual machines across multiple tabs.

The main tab strip is optional.

The main tab strip adheres to the following conventions.

- Minimum number of tabs: There must be at least two tabs on the main tab strip.
- Maximum number of tabs: The recommended maximum is six to seven tabs; this prevents horizontal scrolling as a result of too many tabs, and the growth of text strings from localization or internationalization. (Estimate 30% growth of all text strings due to translations from localization or internationalization).
- Naming of tabs: Tab labels should represent content areas, not action items. They should not combine both content and action.
- Label and Tab color: The text style and color of a selected tab must change to represent the "Selected" state.

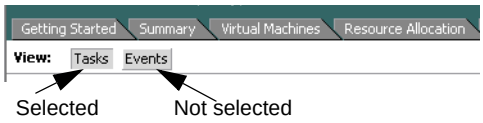
These conventions apply to the main tab strip:

Element	Convention
Tab Text Style	Tahoma Regular, 8pt 11px 0.7em 70%
Tab Selected Background Color	White (#FFFFFF)
Tab Selected Text Color	Black (#000000)
Tab De-selected Background Color	#7B869C
Tab De-selected Text Color	White (#FFFFFF)
Tab Background Area Color	#39516B

Subtab Strip

Subtabs filter the content of the associated main tab. Subtabs only appear when the main tab has sections of content that fall into logical categories associated with a user's workflow goals.

- Minimum Tabs: There must be at least two tabs under a main tab.
- Maximum Tabs: The recommended maximum is six to seven tabs; this prevents horizontal scrolling as a result of too many tabs, and growth of text strings from localization or internationalization. (Estimate 30% growth of all text strings due to translations from localization or internationalization).
- Naming of tabs: Tab labels should represent content areas, not action items. They should not combine both content and action.
- Label and Tab color: The text style and color of a selected tab must change to represent the "Selected" state.



These conventions apply to subtabs:

Element	Convention
Tab Text Style	Tahoma Regular, 8pt 11px 0.7em 70%
Tab Selected Background Color	Blue-gray, # CDD3DA
Tab Selected Border	Top, left one pixel solid. Gray (#9D9DA1)
Tab Selected Text Color	Black (#000000)
Tab De-selected Background Color	Dark grey (#9D9DA1), 2px
Tab De-selected Border	Bottom, right, one pixel solid, Gray (#9D9DA1)
Tab De-selected Text Color	Black (#000000)
Tab Background Area Color	White (#FFFFFF)

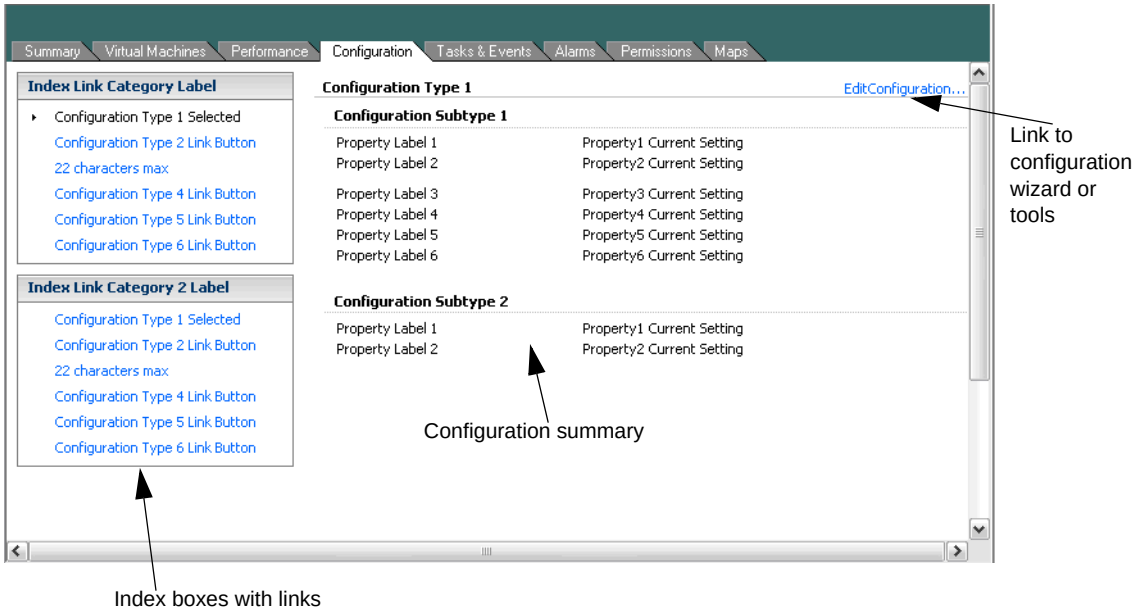
Configuration Tab Page Layout

The **Configuration** tab page presents the configuration options for the selected resource. This tab should consist of index links, summaries, and editing wizards and tools.

The vSphere Client simplifies configuration tasks by displaying only the configuration detail summaries and tools relevant to a particular task. You should not present users with all tasks in one screen. You should use the **Summary** tab provides users with an overarching view of configurations and information related to a view or selected object.

NOTE While not encouraged in the vSphere Client, web plug-ins can display configurable entries in the summary details space.

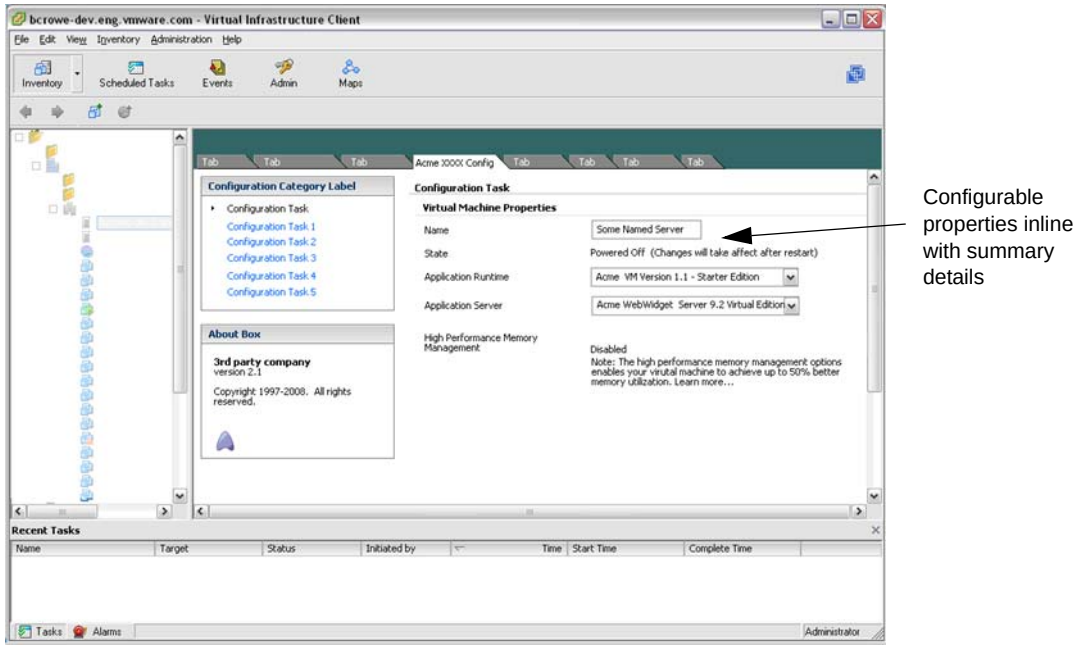
1 User clicks a **Properties** or **Edit** link to open a configuration wizard or widget to edit the configuration.



The interaction for the vSphere Client works as follows:

1 User clicks an index configuration link from the left side.

2 The related configuration summary displays on the right side.



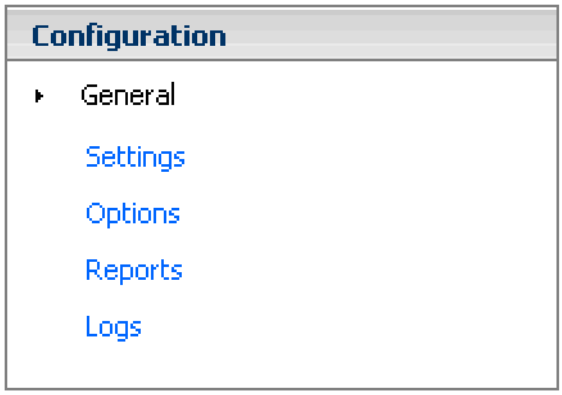
These conventions apply to the configuration summary section.

- Sections can be grouped by a heading title in bold and a horizontal line (line color: #C0C0C0).
- Subsections can be separated by a dotted horizontal line and slightly indented.
- Commands that apply to the content inside the grouping should be placed in links starting aligned from the right of the horizontal line (line color: #0066FF).

Index Boxes

The vSphere Client simplifies complex configuration tasks by grouping similar configuration index links into logical categories within a box called an *index box*. For example, hardware configuration tasks are separated from software configuration tasks.

Index boxes appear on the left side of the **Configuration** tab screen. Clicking an index link in an index box populates the right side of the **Configuration** tab screen with related configuration content.



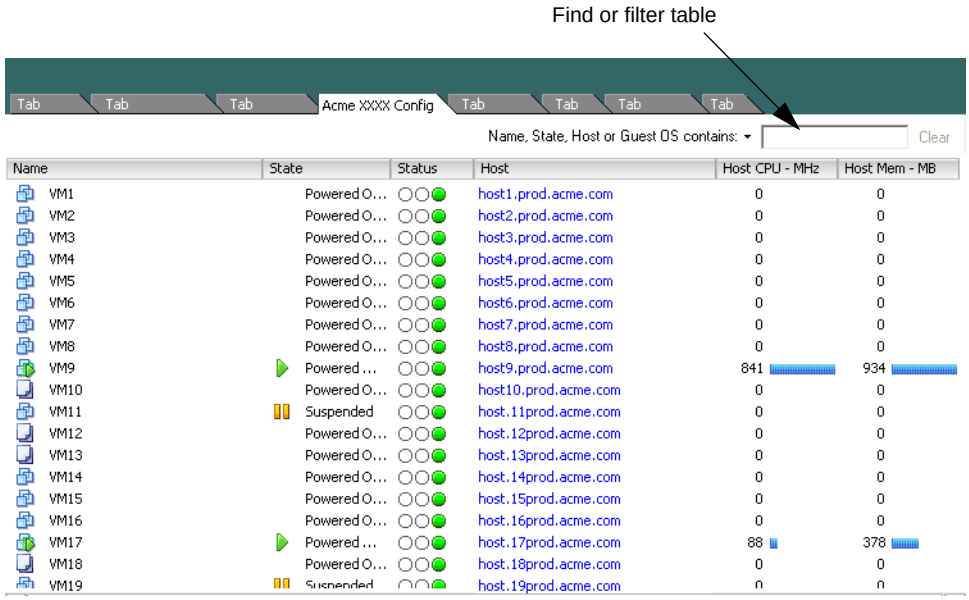
These conventions apply to index boxes:

Element	Convention
Index Box Title Text	Tahoma Bold, title capitalization, 8pt 11px 0.7em 70%
Index Title Box Header	Background gradient ( )
Index Box Title Text Color	#0F3F6D
Index Text List Color Selected	Black (#000000) with a small black arrow icon (‣)
Index Box Text List De-selected Link Color	#0066FF
Index Box Border COlor	1px #7a7a7a
Index Box Background Color	White (#FFFFFF)
Index Box Title Background Color	Gradient Bottom: #E9EAEB Top: #CED0D0

Tables

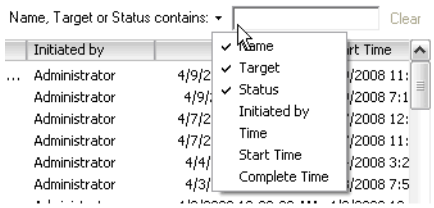
Tables are the primary method for displaying data in the vSphere Client. VMware recommends that all tables displayed in information panes provide read-only data. Inline editing should be confined to configuration views, dialogs, and wizards to minimize accidental editing or changing of configurations.

Tables can allow users to alter the view of the table contents by sorting or filtering columns. Links may be provided in tables to allow users to drill down on table contents or navigate to objects.



The vSphere Client provides a filter widget that allows users to view a subset of the data contained in a table. Users select a set of columns, specify a text string, and press **Enter** to view the subset of data rows that match the specified text string in the selected columns.

The table filter widget consists of a label, a column selector dropdown, a text field and a **Clear** button. The table filter is always aligned to the top right of the table.



These conventions apply to tables:

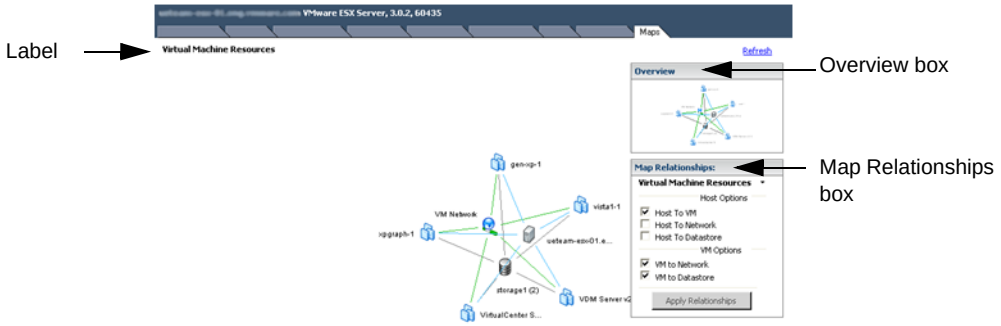
Element	Convention
Table Header Text Style	Tahoma Regular, 8pt 11px 0.7em 70%
Table Header Text Color	Black (#000000)
Table Header Border Line (outer)	Grey (#C0C0C0), 1px
Table Header Border Line (top inner)	White (#FFFFFF), 1px
Table Header Divider Line	Dark grey (#9D9DA1), 2px
Table Header Background Color	Linear gradient: ■ Top #C8C8C8 ■ Bottom #F7F7F7
Table Header Reference Image	
Table Cell Background Color	White (#FFFFFF)
Table Cell Grid Color	White (#FFFFFF)
Table Cell Text Style	Tahoma Regular, 8pt 11px 0.7em 70%
Table Cell Text Link Color	Blue (#0000FF)

Element	Convention
Alignment: Column Headers	Aligned with data
Alignment: Cell Data	Text: Left align Numeric values: Right align for totaling and value comparisons

Maps Tab Pages Layout

These conventions apply to the Maps tab.

Element	Convention
Label	Tahoma Regular, 8pt 11px 0.7em 70%
Overview box	This box allows user to pan to an area or zoom in. This box is always present.
Map Relationships box	This box allows user to view different types of resources, and select particular object type relations within these views. This box is present when relevant.



Wizards and Workflows

A wizard is a sequence of steps that guides the user in completing a task. A wizard minimizes the user knowledge required to complete a task and promotes a smooth, inductive navigational flow.

Wizards use pages, with the task objective placed in the title bar.

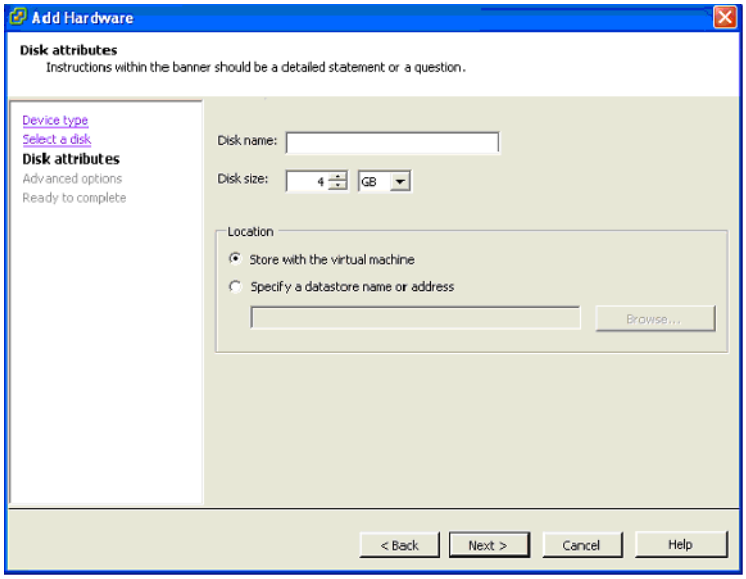
VMware encourages the design of modeless wizards.

Advantages of modeless wizards

- Within the wizard sequence, the user can look up or research custom settings.
- Modeless wizards might allow for an easier transition to a web-based application.
- With a modeless wizard design, the designer needs to consider the possibility of multiple instances of the wizard.
- A modeless wizard must frequently update and verify that the values are valid.

Disadvantages of modeless wizards

If a large number of dependencies exist, then a modal wizard may be preferred.



Wizard Components

Wizards include the following components:

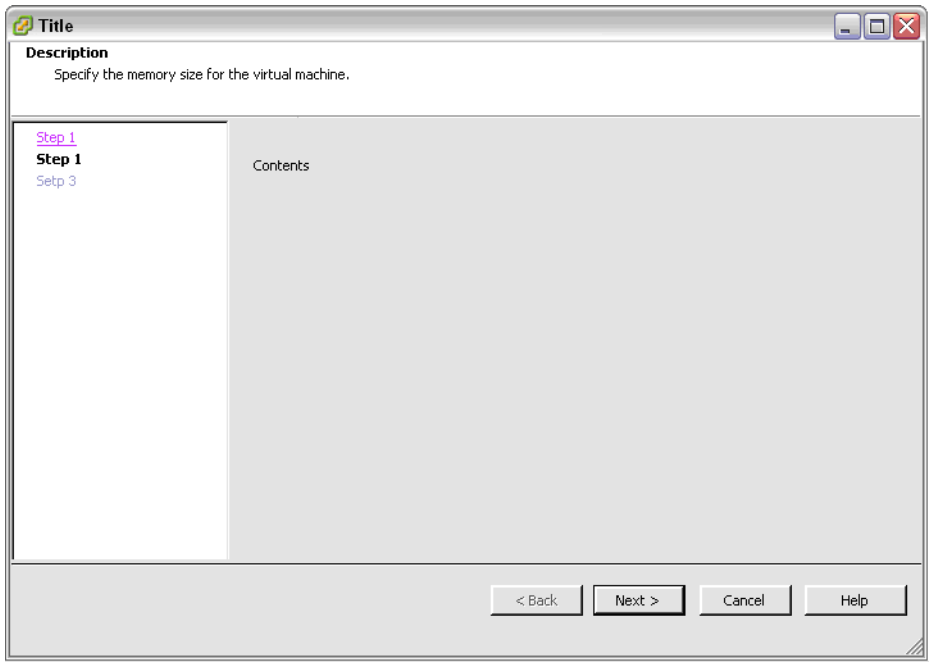
- A title bar that identifies the wizard task objective and title.
- List of steps for completing the overall wizard task on each wizard page. Use bold text to indicate the current step and links to allow the user to navigate to other steps of the workflow.
- At least three pages for gathering information and user selections.

  NOTE If there are only one or two pages within the wizard for the user to make selections, then use dialog boxes rather than a wizard.

- Descriptive titles and objectives for each major step and sub-step.
- Instructions for required decisions and actions to complete each step.
- Each page includes instructions that state the user's objective for that page.
- A centralized editing area with optional text and controls.
- Command buttons to commit to the task or to proceed to the next step. The Back and Next buttons appear on each page of the wizard, with the exception of the final wizard page, in which the Next button is replaced by a Finish button.
- A Summary page is optional and includes selected settings for the user to confirm.
- A Progress page is also optional and includes a progress bar which illustrates the progress of operations that take more than four seconds. Typically, you use a Progress page following a user action that requires a lengthy operation.

  Alternatively, you can present a notification message as feedback to the user during a lengthy operation.
- Within wizard pages, the Next button is the selected default command button, with the exception of the last wizard page.

- Within the last page of a wizard, typically the Summary page, the Finish command button is the selected command button.



Wizard Titles

Create wizard titles by stating the wizard task objective.

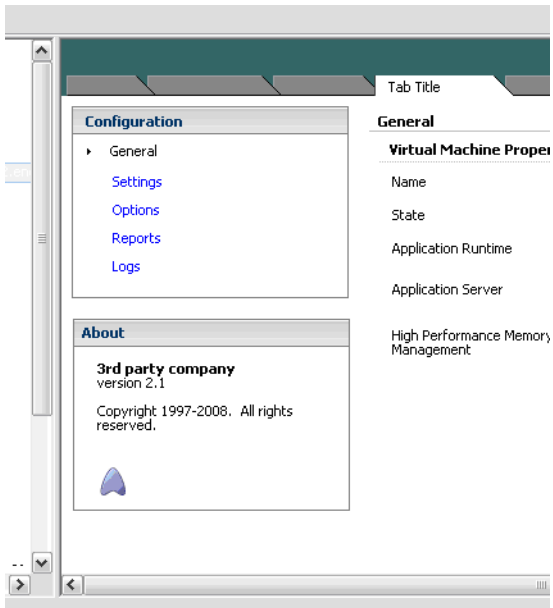
Create a Virtual Machine

Do not use the term *wizard* in the title. However, you can refer to a wizard in the instructional text.

Use the Create a Virtual Machine wizard.

Partner Logo

You can place partner logos in the About box, which is aligned to the bottom left of the information pane. The About box contains branding, name, copyright, version, and other metadata about the plug-in.



Error Reporting

Providing frequent, appropriate, and specific user feedback enhances the ease of use. Also, validating user input confirms the accuracy of the user selections and commands. The following types of feedback to the user are required throughout the user interface.

Alerts

The industry standard is to present feedback to the user in the form of alerts with three levels of criticality. In order of least critical to most critical, the three types of alerts are as follows:

Type	Icon
Notifications	 or 
Warnings	
Alarms	

Progress Indicators

Progress indicators are bars that show the user the progress of a lengthy operation that takes over two seconds. A progress bar communicates to the user the following information:

- An extensive operation has started.
- The operation is in progress.
- The operation has not locked up.

An asynchronous progress indicator informs the user that the operation is occurring; however, it does not provide an estimate of when the operation will be completed.

If the system response latency is greater than two seconds, then provide feedback to the user with an asynchronous progress indicator to show system processing is occurring.

Progressive Escalation

As the feedback to the user becomes more critical, the user interface presentation should become more prominent.

The most critical alarms that require user action should be placed within a modal dialog.

Warning Messages Guidelines

A warning message is used to:

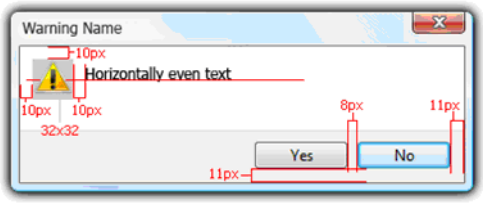
- Inform users of possible consequences for an action. Users may be instructed to confirm or cancel an action.
- Display a warning that informs the user of a potentially dangerous outcome.
- State unexpected results, such as "Graph cannot be displayed".
- State unexpected situations or system conditions, such as CPU incompatibility.

When you present a warning, adhere to these conventions:

- Display the Warning icon.
- Get to the point of the condition; advanced users frequently skim the messages.
- State the message objective in the title, for example, "Confirm..."
- Do not start the message with "Are you sure you want to…"

- State the object by name, if known. For example, "Delete virtual machine XYZ?"
- State the consequences for the action (Current state of the virtual machine will be lost unless it has been saved in a snapshot) along with the confirmation question (Revert to snapshot TEST?).
- Display a warning that informs the user of a potentially dangerous outcome and suggest an alternative user action.

The following warning message dialog box states the type of warning in the title bar by using title capitalization. Also, the Yes and No command buttons are right justified in the lower right of the window. The warning text is aligned horizontally with the vertical center of the warning icon:



Error Message Guidelines

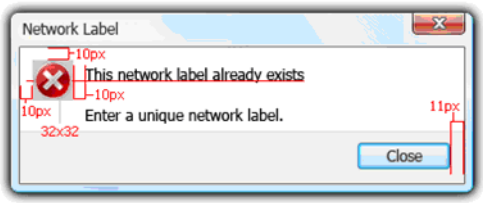
Error messages should state the specific error and the recommended user actions to correct the error. Since users do not regard errors as an “OK” condition, you should communicate a positive and to use the Close command button in error messages.

When you present an error, adhere to these conventions:

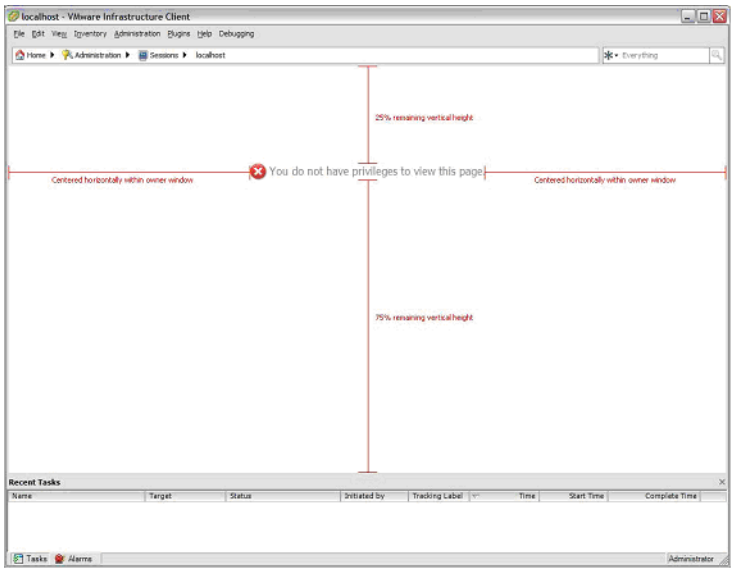
- Use the same error icon for all error messages.
- State the type of error in the title bar of the error message, using title capitalization.
- State the error condition in one sentence, and provide a solution for the error in a second sentence.

  If there is insufficient screen space for two sentences in the error message, one stating the error and one stating the corrective action, give priority to the sentence providing the user with corrective action. It is more important that the user be given a suggestion for a corrective action, so the user could proceed.
- Separate the error from the solution by using a blank white space.
- Align the top of the text capital letters horizontally with the top of the white "X" inside the error icon.
- Place the Close command button in the lower right corner of the error message window.

In the first error example, the layout specifies the pixel measurements between the 32x32 icon box and the left edge of the error message text.



In the second error example, the error message text is placed within the owner window with 25% remaining vertical height above the error message, and 75% remaining vertical height below the error message, as well as being horizontally centered.



Best Practices

Top 10 UI Design Tips

The Top 10 UI Design Tips provide a summary of UI guidelines to apply to develop a compelling and intuitive user interface:

1 Design for users and their specific tasks.

2 Emphasize solutions that integrate features. Ensure every feature benefits the user.

3 Provide specific, relevant user feedback.

4 Automate common tasks and provide intelligent defaults.

5 Manage complexity by using progressive disclosure to present only information the users need.

6 Present one concept at a time.

7 Follow UI design standards and guidelines for consistency.

8 Stability and predictability—Ensure the same actions should produce the same results.

9 Adaptability—Provide user control and short-cuts.

10 Identify user errors and suggest actions to correct the errors.

Excerpted from VMware UI Design Standards and Guidelines 5-5-08, page 5.

Additional Tips for Plug-in Developers

- Seamless integration prevents a jarring experience. Embed the main UI inside the vSphere Client framework. Follow vSphere Client conventions to take advantage of users' past experiences. Don't force customers to learn a new model.
- Use a wizard for any task consisting of several steps that must be completed in a specific order. Think twice before embedding the wizard in a tab. Embedded wizards can cause the user to lose context. For more wizard guidelines, see "Wizards and Workflows" on page 10.

- Make windows resizable. Allow contents to be viewable for resolutions as low as 800 X 600px. Content should work well at 1024 X 768px. Dialogs should be no larger than 750 X 550px.
- Performance. For queries on lots of data, avoid displaying all information for all levels up front. Support drill-down behavior. Provide a filtering mechanism for long lists of items.
- Use scroll containers sparingly. Avoid nested scroll containers.
- Minimize accidental editing. Require explicit actions to edit, avoid inline editing off of the main application. Use dialogs or confirmations for configuration changes.
- Keep system status visible, particularly on long, on-going tasks.
- Consider accessibility, such as 508 compliance. Make the UI keyboard-accessible.
- Consider a localization strategy for the text and images within each plug-in. Separate all resources from the source code of a plug-in itself so these resources can be translated to a new locale.
- Use understandable messages to help users recover from error conditions.