Pill Based Controls

includes Target Elements, Launch Buttons, and Auto-Complete

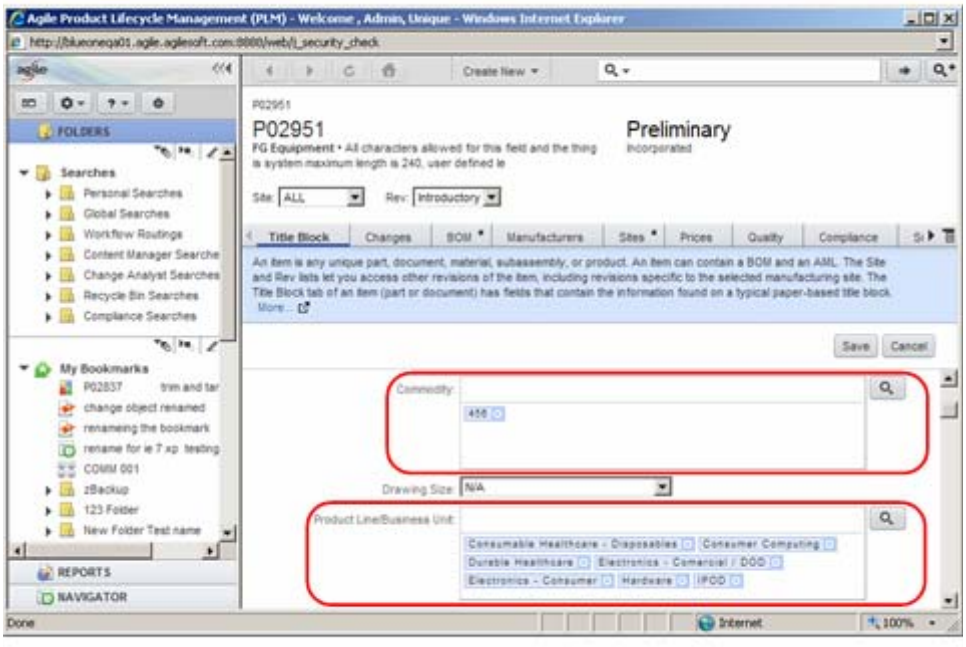
In many places within the Agile Application, the value of a form element or table cell will be a list containing one or more entries where each entry corresponds to a record in the database. These records will represent business objects or entries in administratively defined lists.

Business objects (e.g. Customers, Parts, Manufacturers, Sites, Programs, etc.) are derived from Agile classes or base classes. Records of this type are dynamically defined during the course of operating the business.

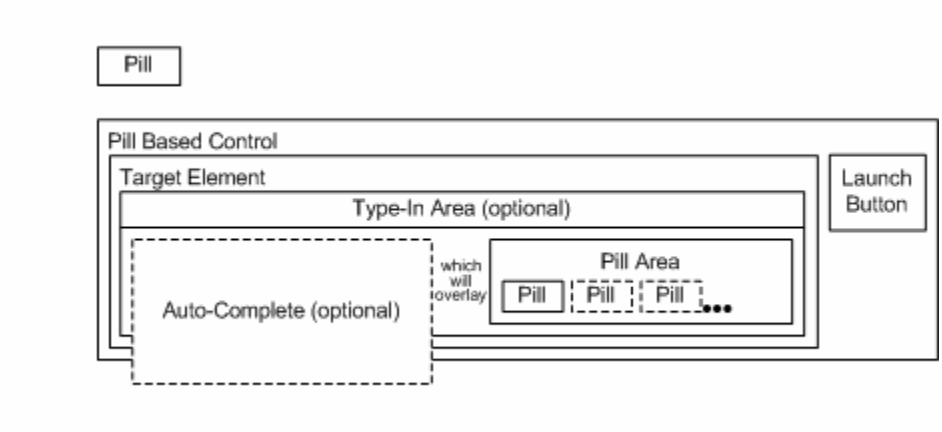
Administratively defined lists (e.g. Countries, Currencies, Priorities, etc.) are statically defined during the process of installing and configuring Agile PLM although they can be changed by the administrator over time. The term non-object data refers to entries in this type of list.

To ease the data entry burden of assigning these records to target elements, the Agile User Interface encapsulates each record into pill format as it is assigned. These pills can later be selected and deleted by the user.

Element Specification



Pill Based Control Building Blocks



Pill based controls include a Target Element, a Palette Launch Button, and a Palette (not shown). The Target Element always contains a Pill Area and generally contains a Type-In Area supported by an Auto-Complete mechanism. Please note that the Auto-Complete component floats above the plane on which the other control elements are arranged. Thus it is not limited by the geometry of Target Element or the control. When open, the Auto-Complete component covers all or part of the Pill Area.

Which of these building blocks are present, what form they take, and which blocks are excluded is determined by the type of the control.

This is a generic specification wherein each of these building blocks, and every form each building block can take, is specified in detail in the sections that follow.

The combination of these building blocks into specific controls will be discussed in separate UI specifications for each individual control.

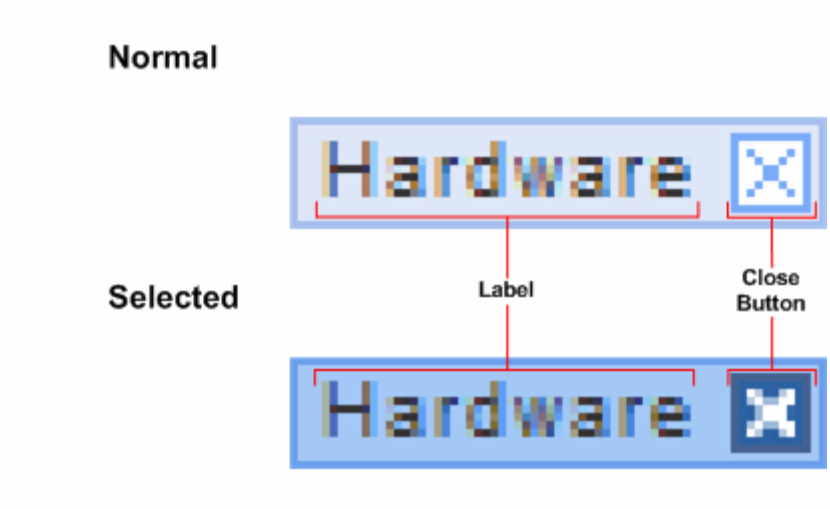
A Note on Layout

Conventions

- Borders, unless otherwise noted, are 1px wide and not dimensioned by interval brackets on illustrations.
- Dimensions of components as illustrated include their borders (if any).
- Dimensions of the space between components do not include the borders of the components.

Pills

Anatomy



Behavior

Pill Labels and Tooltips comply with the *Naming and Tooltips* UI Specification except as noted below.

Pill Label Truncation

- The maximum length of a pill label is determined by the width of the Target Element. If the label in a Pill is longer than the maximum length of the Pill, the text will be displayed left justified and truncated with trailing ellipsis to mark the truncation.

Tooltips

- In general, the label in a pill will reflect the full name of object or non-object list entry sufficient to uniquely identify the data associated with the pill. There are two important exceptions.
- The first is when the pill label is truncated as described above. The tooltip for a pill with a truncated label will be the full text of the label without truncation.
- The second is when the data comes from a tree (AKA cascade) structured data set. For example, a pill with the label “San Jose” could identify either:

  USA | California | San Jose

  Costa Rica | San Jose

  The tooltip for pills from these structured data sources will be the fully qualified unique identity of the pill in the form shown above.
- All other pills will not display a tooltip.

Mouse Buttons (on Pill Label)

- Click — Select the Pill.
- Control-Click — Toggle the selection of the Pill.

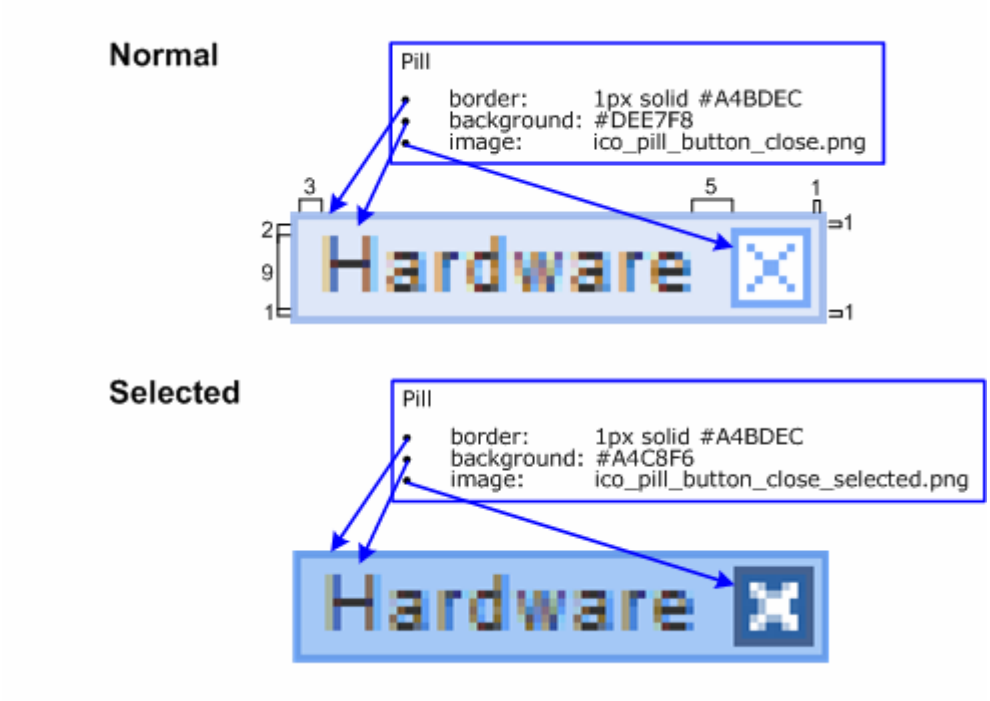
Mouse Buttons (on ☒ Button)

- Click — Remove Pill from Pill Area.

Mouse Over

- Cursor: — pointer;
- Tooltip — as described above.

Layout, Appearance and Styling



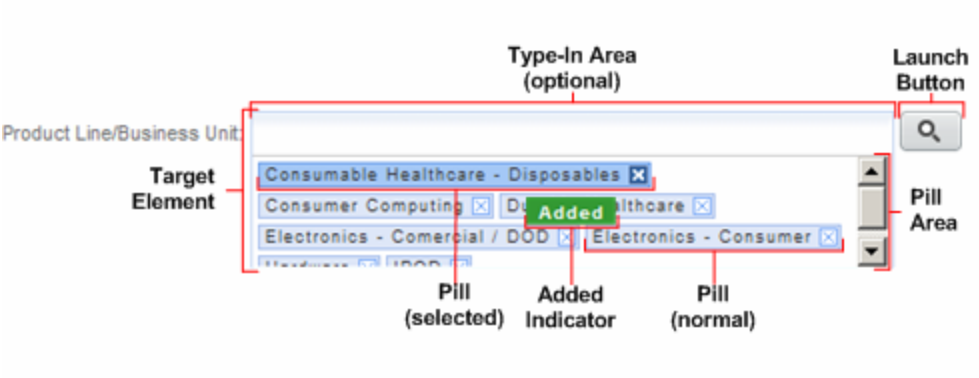
The Pill label is rendered in Agile Normal Font except for:

- font-size — 9 px
- letter-spacing — 1.06667 px
- line-height — 12 px

Please see the *Fonts* UI Specification for the definition of the Normal Font.

Pill Based Editable Form Control

Anatomy



Behavior

Common Behaviors

Launch the Palette / Re-Purpose the Palette

Please note that *Launch the Palette* will *Re-Purpose the Palette* if it is already open.

Please see the *Palette* UI Specification for the detailed definition of these behaviors.

"List Too Long" Error Recovery

Attributes with a Display Type of List (vs. Search) will be serviced by a filter based control and the entire contents of the List will by downloaded to the client browser as the page is loaded. This works well as long as the List is short (no more than 500 values).

If the number of entries in the List exceeds 500, the control will automatically be converted to a search based control (by changing the Display Type to Search) when the List is loaded.

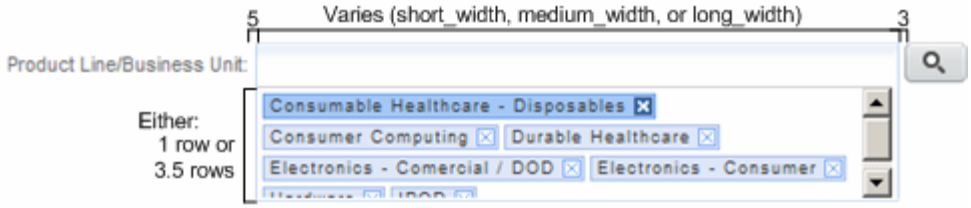
Thus, while the appearance of the control will not change – particularly the icon on the Palette Launch Button – the behavior will be that of a search control.

The modified behavior is:

- The Launch Button will launch a search palette (see the Palette UI Specification).
- Auto-Complete will be restricted as required to maintain the performance of search controls. The likeliest form of this restriction is a delay in the appearance of the suggestions list until several characters have been typed.

On subsequent references to the List, a search based control will be created.

Layout, Appearance and Styling



Please see the sections that follow for specific information about the Target Element, Type-In Area, Auto-Complete List (not shown), Pill Area and Launch Button subcomponents identified in the illustration above.

Target Element

A Note on Focus

The UI controls that use Pills are made up of tightly interconnected UI elements. In simple UI elements, the user's attention is focused on the element and the user's actions apply only to that element.

In these controls, actions performed on one element may result in apparent behaviors within other elements even when there is no apparent behavior at the point where the action was initiated.

Within these controls, only two elements can hold the focus. These are the Target Element and the Launch Button. These two elements are included in the tab order.

The Type-In Area, Auto-Complete List, Pill Area and the Pills cannot hold the focus. Clicking on these elements or using keyboard shortcuts that act upon these elements will not shift the focus away from the Target Element. These elements are not included in the tab order.

All keyboard input is directed at the Target Element. This input will trigger behaviors. These behaviors will be documented in the element where the behavior occurs rather than at the high level of the Target Element.

Behavior

Common Target Element Behaviors

Pill based controls support either Single Select or Multi-Select modes.

Auto-Complete (if and only if Type-In / Auto-Complete are present)

- Put printable characters, as they are typed, into the Type-in Area.
- Launch Auto-Complete List on first typed character.
- Further constrain the Auto-Complete List with each added character. The Auto-Complete feature uses a "starts with" filtering strategy to narrow down the list.
- When the Auto-Complete list has been sufficiently constrained, the user may accept a single Auto-Complete suggestion or choose from a short list of possible values. How this is done is described below.
- The user may type (or paste) multiple entries separated by semi-colons. Auto-Complete will apply only to the rightmost entry in the Type-in Area.

Flash the Added Visual Indicator

- Flash the *Added* visual indicator by:
  - Displaying it at full brightness immediately.
  - Holding it at full brightness until all Pill values have been created and assigned to the Target Element, or for at least 0.25 seconds, whichever is longer.
  - Fading it away over 0.4 seconds.

Convert the Type-In Area to Pill Format

For Single Select Controls

- If the Type-In Area contains a single valid entry, convert the entry to Pill format.
- If not, the Type-In area is invalid in its entirety.
- Leave all invalid data in the Type-in Area and display it in the Agile Bold Error font.

For Multi-Select Controls

- Each element in a semi-colon separated list will be validated independently.
- Convert all valid entries to Pill format.
- Remove all valid data from the Type-in Area.
- Leave all invalid data in the Type-in Area and display it in the Agile Bold Error font.

Add the Pill(s) to the Pill Area

For Single Select Controls

- Replace the current Pill with the single new Pill value.
- If multiple values are dropped on the Target Element (from a source other than the Palette which enforces the single value restriction), the data will be used to populate the Type-In Area with a semi-colon separated list which will be processed as documented above in *Convert the Type-In Area to Pill Format*.

  Controls that lack a Type-In Area (please see the Pill Based Editable Cascade Control section near the end of this specification) can only receive data from the Palette which guarantees success.

For Multi-Select Controls

- Insert the Pill(s) at the front of the Pill Area. If a value is already in the list, it will not be moved to the front.

Validate the Data (if and only if Type-In / Auto-Complete are present)

- Close the Auto-Complete List if open. Please note that data from the Auto-Complete List will always validate correctly.
- *Convert the Type-In Area to Pill Format*.
- *Add the Pill(s) to the Pill Area.*
- *Flash the Added Visual Indicator* in the Target Element to notify the user that the value(s) that have disappeared from the Type-in Area have been added to the Pill Area. If the value(s) to be added are already in the list, they will be left in their original position(s) but the *Added* indicator will flash to indicate their presence.

Receive Data from the Palette

- Convert Palette Data to Pill format. Please note that data from the Palette will always validate correctly.
- *Add the Pill(s) to the Pill Area*.
- *Flash the Added Visual Indicator* in the Target Element (on drop) or in the Palette (all other cases) as appropriate.

Gains Focus

- Show focus ring (around entire Target Element).
- Display the type-in cursor in the Type-in Area (if present).
- If the user tabbed into the Target Element, highlight any text in the Type-in Area for deletion or replacement. [xxx Stopped working 9/6/07]

For Single Select Controls Only

- Unconditionally highlight any text in the Type-in Area for deletion or replacement.
- Select the Pill (if any).

For Cascade List and Multi-Cascade List Only

- Select the Pill (or first Pill for Multi-Cascade List) if present.
- If the Pill Area is empty, shift the focus to the Launch Button.

- *Re-Purpose the Palette* (if open).

Loses Focus

- Close the Auto-Complete List (if present and open).
- *Validate the Data*.
- Clear the selection on all Pills.
- Remove the focus ring.

Keyboard Data Entry

- Collect characters and perform the *Auto-Complete* behavior on the result.

Copy

- Text displayed anywhere in the Target Element may be selected and copied, in whole or in part, to the clipboard. Selecting a Pill does not select the text. The standard “drag across desired text” paradigm applies.

Keyboard Controls and Shortcuts

- Enter — *Validate the Data*.

  The focus remains on the Target Element.

- Tab (Shift-Tab) — *Validate the Data*.

  Shift the focus away from Target Element and to the next (prior) UI control in tab order. The next control will be the launch button.

- Esc — Leave the value of the currently highlighted row from the Auto-Complete List in the Type-in Area.
  Dismiss the Auto-Complete List.

- Ctrl-L — *Launch the Palette*.

The following keyboard controls are typed within the Target Element but the behaviors that they trigger occur in the UI elements listed below.

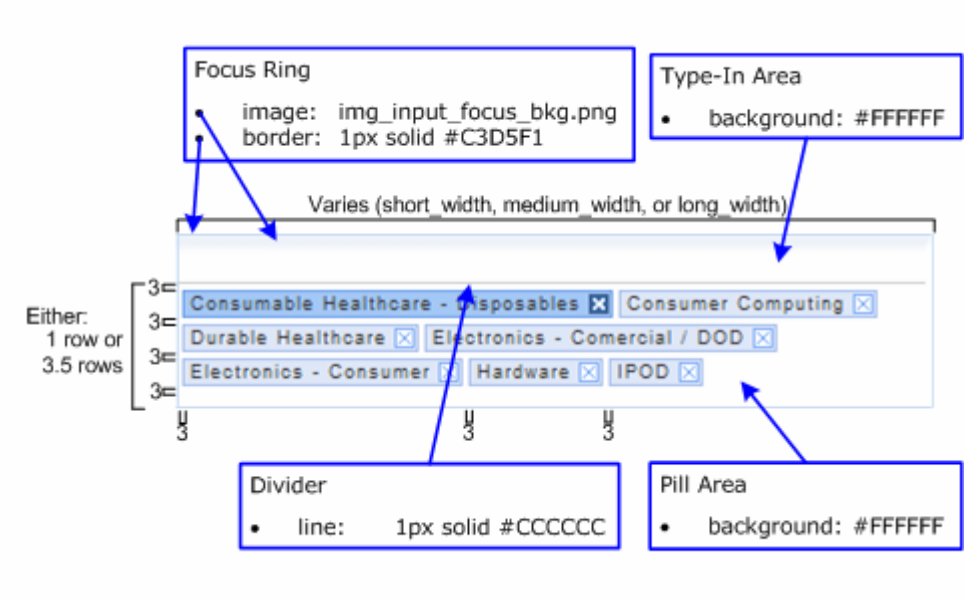
- See Type-In Area
  - Down-Arrow
  - Left-Arrow / Right Arrow — if and only if there is text in the Text Area
  - Delete — if and only if there is text in the Text Area

- See Auto-Complete List
  - Up-Arrow
  - Down-Arrow
  - Right-Arrow

- See Pill Area
  - Control-A
  - Delete — if and only if the Text Area is empty
  - Left-Arrow / Right-Arrow — if and only if the Text Area is empty
  - Shift-Up-Arrow
  - Shift-Down-Arrow
  - Shift-Left-Arrow
  - Shift-Right-Arrow
  - Shift-Home
  - Shift-End

The complete specification of these UI elements can be found after the next page which completes the high-level specification of the Target Element with a discussion of its Layout, Appearance and Styling.

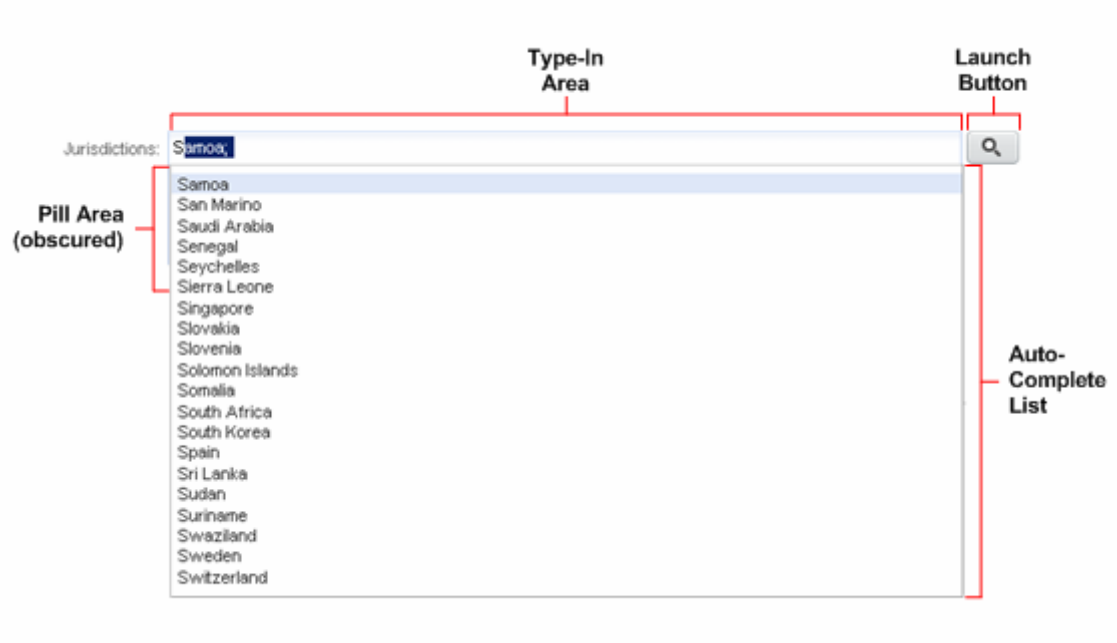
Layout Appearance and Styling



Type-In Area / Auto-Complete (optional)

Pill based controls that allow data to be typed into them will include a Type-In Area that is intimately linked to an Auto-Complete feature.

Anatomy



Type-In Area

The Type-In Area is present in all filter and search based controls. It is not present in cascade based controls.

Except where otherwise noted, the Type-In Area conforms to the *Text Control* UI Specification.

Behavior

The Type-in Area complies with the UI Control Commonalities Specification.

Loses Focus

- *Validate the Data*

Keyboard Data Entry

• Characters Allowed	Any characters within the character set defined for a data element are permitted.
• Semi-Colon	For lists allowing multiple values, semi-colons are used as separators between values. For data elements that can only accept one value, semi-colons are not treated as special characters.
• Error Indication	As long as the last value in the Type-In Area matches one or more Auto-Complete suggestions, the Type-In Area will be displayed in the Agile Normal Font. When the last value is invalid, the entire Type-In Area will be displayed in the Agile Bold Error Font.

Keyboard Controls and Shortcuts

• Down-Arrow (If the Auto-Complete List is …		
	… Open) then	Highlight the next row in the list.
	… Closed) then	Select the first Pill.

If and Only If the Type-In Area Contains Characters	
o Delete	Standard browser behavior (delete the character to the right of the text insertion point).
o Left-Arrow / Right-Arrow	Standard browser behavior (move cursor).

Paste

• Text data	If multi-line data is pasted; only the first line is accepted. Subsequent characters are ignored.
• Non-text data	Ignored.

Auto-Complete List

The Auto-Complete List is present in all filter and search based controls. It is not present in cascade based controls.

Behavior

The Auto-Complete List will appear as soon as the user starts typing in the Type-In Area.

This may change to defer A-C for search controls until the user types several characters for performance reasons.

As the user continues to type, the list will be reduced. The remaining characters of the first suggestion will be placed in the Type-In Area to the right of the text insertion point and highlighted for replacement.

At any point, the user may click on a suggestion, or use the down arrow keys to navigate through the Auto-Complete List.

The complete list of user controls follows.

Keyboard Controls and Shortcuts

• Enter	Accept the currently highlighted suggestion from the Auto-Complete List. *Validate the Data*.
• Right Arrow	Accept the highlighted suggestion and close the Auto-Complete List. If multiple selections are allowed by the control, invite the user to type additional entries by appending a semi-colon (actually "; ") to the text in the Type-In Area to separate the entries.

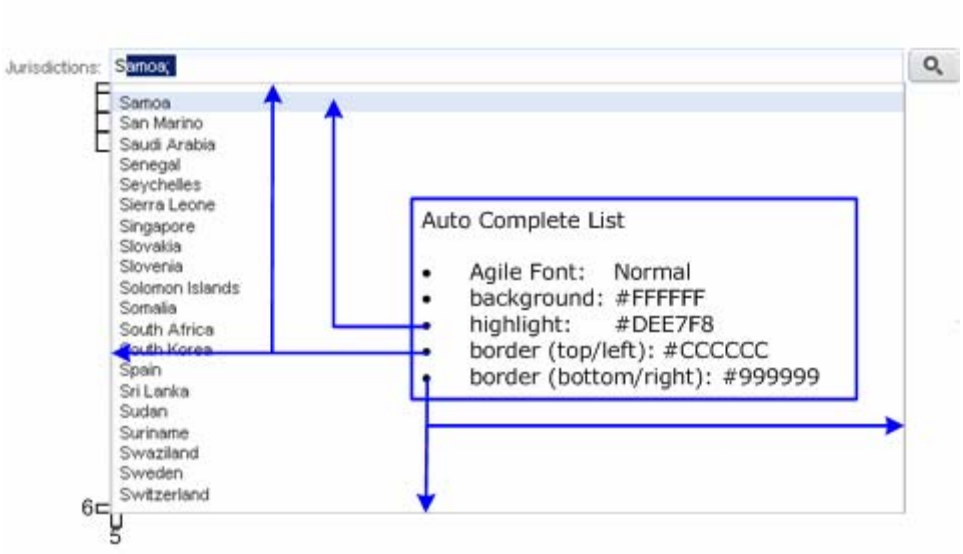
• Down Arrow	Move highlight one row down in the Auto-Complete List. Update the Type-In Area to reflect the value of the highlighted row. At the bottom of the list, do nothing.
• Up Arrow	Move the highlight one row up in the Auto-Complete List. Update the Type-In Area to reflect the value of the highlighted row. At the top of the list, remove the highlight from the Auto-Complete List and revert the Type-in Area to the leading character(s) as typed by the user. Keep the Auto-Complete List open (a second Up Arrow will close it).

Mouse Buttons

• Click (on a row)	Accept the suggestion from the Auto-Complete List. *Validate the Data*.

Layout Appearance and Styling

The Type Ahead Suggestions list is limited to 20 items max.

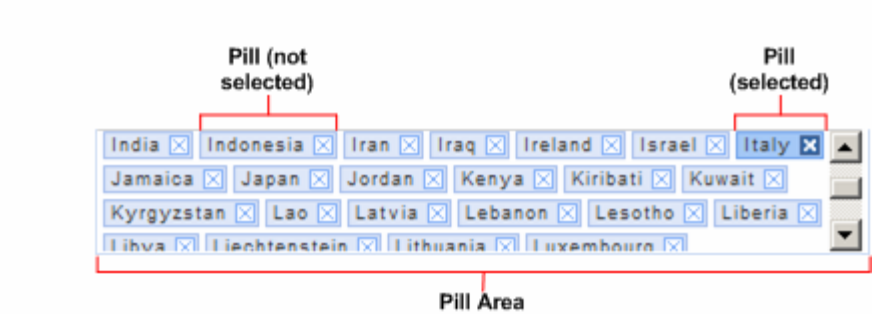


Pill Area

Depending on the control, the Pill Area may contain either a single Pill value or a list of values.

Lists of Pills may be of any length and will wrap so that they can be read left-to-right, then top-to-bottom. Lists are scrollable if they grow beyond the ability of the Pill Area to fully contain them.

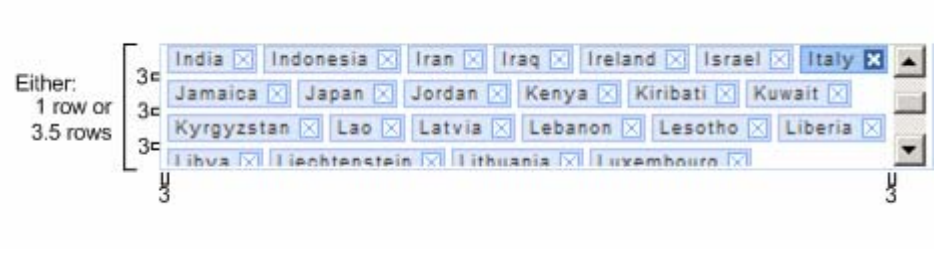
Anatomy



Behavior

The behavior of the Pill Area is documented below under Pill Area (Single Select) and Pill Area (Multi-Select).

Layout Appearance and Styling



Pill Area

- height: 22 pixels (1 row) for Single Select Controls
  63 pixels (3.5 rows) for Multi-Select Controls

Pill Area (Single Select)

Behavior

Gains Focus

- Select the Pill.

Loses Focus

- Clear the selection of the Pill.

Keyboard Controls and Shortcuts

	If and Only If the Type-In Area is Empty
• Delete	Remove the selected Pill from the Pill Area.

Mouse Buttons (on ☒ Button)

• Click	Remove Pill from Pill Area.

Mouse Over

• Cursor:	pointer;
• Tool Tip	None on the Pill Area proper. A Pill may have a tooltip as specified above.

Drag and Drop

• Drop	*Receive Data from the Palette.* *Flash the Added Visual Indicator* in the Target Element to notify the user that a new Pill has replaced the old value in the Pill Area.

Please see the *Palette* UI Specification for more information about Drag and Drop.

This changes when you can drag from other sources. Get this right. According to Michele, the likeliest process is that dragging from non-palette sources will populate the Type-In Area which when validated will report errors. Errors in the Type-In Area imply (to me at least) "that if I only spelled it right, it would work" when in fact the error could much more fundamental – "I just dragged completely wrong stuff to this element". Watch this space.

Pill Area (Multi-Select)

Behavior

Pill Areas that support multiple selections are an extension to the Single Select implementation and are fully compliant with the above specification with the exceptions and additions listed below.

Pills that were in the list when the form was loaded are listed in alphabetic order. As Pills are added, they are added to the front of the list to make the addition more visible to the user. The list is resorted when the page is saved.

Gains Focus

- Do nothing.

  In Single Select mode we make the assumption that the intent of focusing on the Target Element is to change (i.e. delete or replace) the existing Pill value. We facilitate that by automatically selecting the Pill. No such assumption can be made about gaining focus in Multi-Select mode.

Loses Focus

- Clear the selection of all Pills.

Keyboard Controls and Shortcuts

• Down-Arrow	Select the first Pill and clear the selection of all other Pills.
• Control-A	Select all the Pills.

If and Only if the Type-In Area is Empty

o Delete	Remove all selected Pills from the Pill Area.
o Left-Arrow	Select the previous Pill in the list and clear the selection of all others (e.g. walk up the list).
	Scroll as needed to make this Pill visible.
o Right-Arrow	Select the next Pill in the list and clear the selection of all others (e.g. walk down the list).
	Scroll as needed to make this Pill visible.
o Shift-Left-Arrow	Select the previous Pill in the list (relative the previously selected Pill). Retain the selection of other Pills.
	Scroll as needed to make this Pill visible.
o Shift-Right-Arrow	Select the next Pill in the list (relative the previously selected Pill). Retain the selection of other Pills.
	Scroll as needed to make this Pill visible.

• Shift-Home	Select all Pills from the beginning of the list to the last selected Pill in the list.
• Shift-Up-Arrow	ditto
• Shift-End	Select all Pills from the first selected Pill in the list to the end of the list.
• Shift-Down-Arrow	ditto

Mouse Buttons (on Pill Label)

• Click	Select the Pill. Clear the selection of all other Pills.
• Shift-Click	Select the Pill and the range of other Pills between this Pill and the previously selected Pill.
• Control-Click	Toggle the selection state of the Pill. Retain the selection of any other Pills.

Mouse Buttons (on ☒ Button)

• Click	Remove the Pill from Pill Area. Retain the selection of any other Pills.

Drag and Drop

• Drop	*Receive Data from the Palette.* *Flash the Added Visual Indicator* in the Target Element to notify the user that the new Pills have been added to the Pill Area.

Please see the *Palette* UI Specification for more information about Drag and Drop.

Launch Button

Except as noted below, the Launch Button complies with the *Buttons* UI Specification.

Behavior

Mouse Buttons

- Click — *Launch the Palette*.

Mouse Over

- Tool Tip — "Launch the Palette (Ctrl+L)"

Keyboard Controls and Shortcuts

- Enter — *Launch the Palette*.
- Ctrl-L — Launch the Palette.

Please see the *Palette* UI Specification for details about palettes.

Layout, Appearance and Styling

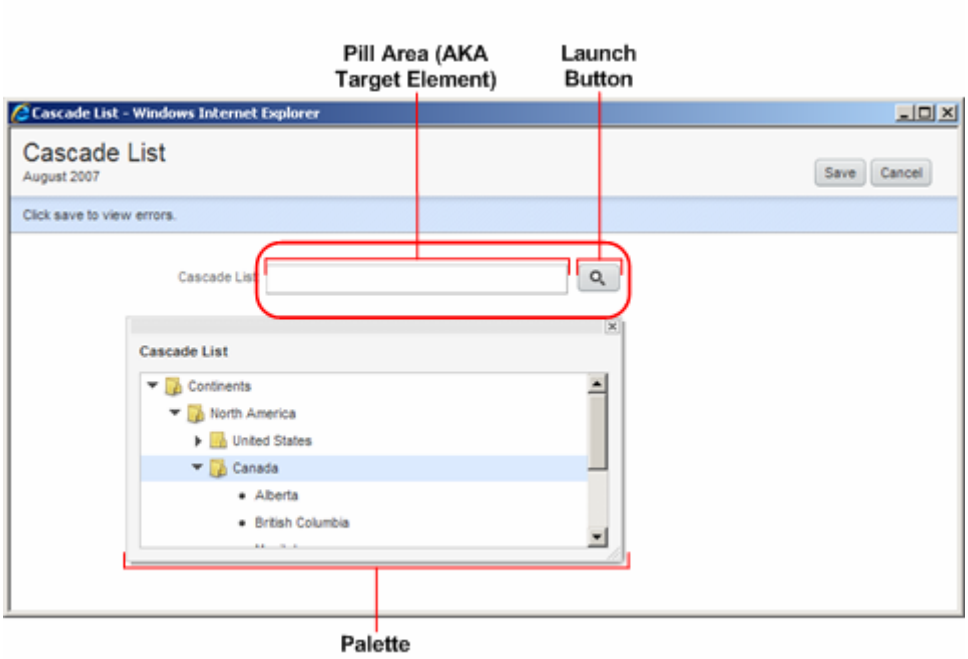
- icon-image —  ico_address_book.png

 ico_view_details.png

The creation of a new icon to fully differentiate between filter based controls, search based controls, and user/user group based controls is under consideration.

Should there be a fourth icon for cascade based controls?

Pill Based Editable Cascade Control



The Cascade List and Multi-Select Cascade List controls appear in forms and tables as a Pill Area and a Launch Button because the only way to enter data into these controls is to browse through a tree structured data set presented in the Palette and select one or more rows from the Palette.

Please see the *Palette* UI Specification (especially the Tree (Cascade) Presentation section) for detailed information about the type of Palette used by these controls.

The behavior, layout, appearance and styling of the Pill Area, Target Elements and Launch Button are compliant with the specification of these elements in the Pill Based Editable Form Control section of this document except for the following:

- Type-In and Auto-Complete are not supported.
- The Target Element and the Pill Area are one. The Target Element can hold focus and is in the tab order.
- When the Pill Area is empty, the focus will automatically shift from the Target Element to the Launch Button because the Palette is the only source for data.
- Drag sources other than the palette allow users to drop values into any Target Element of any control that includes a Type-in Area. The Cascade and Multi-Select Cascade do not have a Type-In Area therefore the only permitted data source will be the Palette (which guarantees a successful drop) and all discussions of error recovery from bad drops in the above section do not apply to these controls.
- When the Target Element gains focus, it will display the focus ring. Any references to the Type-In Area, cursor and highlighted text are not applicable to these controls.
- The first Pill will be selected when the Target Element gains focus.
- Without a Type-In Area, there is no place where data that has been entered but not yet converted to Pill form can exist. Thus all references to validating this data or retaining it for future use do not apply.

Issues

1) The base behavior of the UI elements that make up these complex composite controls differ by platform (OS / Browser combination). These variations are accepted as compliant with this specification.
2) Alternate icons to differentiate non-addressbook controls are under consideration.
3) Unresolved Issue: Alternate Drag Sources (see below).
4) New style added indicator via pulsing the pill coloration is not yet specified.

Alternate drag sources [8/9/07 MY Answers]

The proposal to support drag and drop from Bookmarks and Recently Visited in addition to the Filter or Search Palette opens up some validation issues. At this point in the development process, the Palette changes when the focus changes from UI control to UI control such that if the control expects widgets, the Palette provides widgets and if the control expects gadgets, the Palette provides gadgets. Thus the data is always valid.

How this applies to Bookmarks and Recently Visited opens up the following questions:

1) Can the user drag widgets toward the Target Area of a control that expects gadgets? [Yes]
2) If no, how is this prohibition enforced? [It's not]
3) If yes, can the user drop widgets where gadgets are expected? [Yes]
4) If no, how is this prohibition enforced? [It's not]
5) If yes, what happens during validation? [Text from the Dropped objects will populate the Type-In Area which will then be validated.]
6) If validation is attempted, what happens if a widget has the same name as a gadget? [The name matches and the gadget is Pilled.]
7) Is this possible? [Yes]
8) Is validation by accident acceptable? [Yes]

Change History

November 26, 2007	Vern McGeorge	Revised erroneous image file names.
November 21, 2007	Vern McGeorge	Revised to reflect automatic conversion of filter controls to search controls if the number of list elements exceeds 500.
November 20, 2007	Vern McGeorge	Revised to comply with image file naming convention.
November 6, 2007	Vern McGeorge	All sprint 2.6 comments except Kanda's in.
October 8, 2007	Vern McGeorge	Published for final review.
September 6, 2007	Vern McGeorge	Published for review by Michele.
August 9, 2007	Vern McGeorge	Reorganized and published for review.
June 12, 2007	Vern McGeorge	Revised during clean-up of Multi-Select list.

May 30, 2007	Vern McGeorge	Re-factored out of Multi-Select List and Single-Select Search specifications.