velocity
FROM SCHWAB
SIGNATURE SERVICES

Building Velocity

INTERFACE DESIGN STYLE GUIDE

CRS Interactive
Version 1
October 12, 1999
Confidential to Charles Schwab & Co., Inc.
[URL FOR STYLE GUIDE WEBSITE]

Contents

1 *Building Velocity: an overview*

1.1 Introduction

Velocity Trading Desktop uses the latest technology to bring together the power of the Internet and the order-management efficiency of desktop software. This innovative new tool, designed especially for our active traders, allows the user to develop and implement complex equity trading strategies.

Velocity incorporates Marimba's Castanet products, affording its users ultimate access to Schwab's online trading facilities. Schwab's active traders enjoy all the tools necessary to manage multiple accounts, enhanced by the latest internet technology.

1.2 Velocity 2.x design principles

The primary objective of the user interface design for Velocity 2.x is to facilitate the rapid deployment of a Java-based trading tool. The selection of controls and display components has been restricted as a means of reducing scope and purposefully forcing reuse in the interface development.

The fundamental parameters set for Velocity 2.x user interface design are:

- use of containers to establish the program environment and workspaces
- use of tabs for navigation
- exclusive use of the Swing Set of graphical user interface components (buttons, grids, text displays, form controls) in its current version
- use of existing data and APIs (primarily, but not exclusively from Schwab's Web Site)

Designs built for Velocity 2.x are meant to provide users with easily understood information and access to related functionality. Order and emphasis of components on screen are driven by functional priority. No training or support material can be assumed to help the user navigate the application or a given operation. In general, the designs avoid hidden functions or non-text labels.

Evolution of the design for Velocity 2.x takes place through a process of additive enhancements to the programs primary areas (Account Summary, Account Detail, Orders, Quotes, Setup and New Trade). The design does not preclude additions which may require new spaces, but does not readily facilitate enhancements of this scale. From a design perspective, the Velocity 2.x interface will grow obsolete as the need for multiple workspaces and their complimentary functionality become part of the application requirement, and/or when the primary objective of the interface includes or is metered by inclusion of a brand or proprietary methodology.

1.3 Purpose and Use of the Velocity Style Guide

The Velocity Style Guide is a tool intended to establish practical guidelines for the development of the Velocity User Interface. Basic information about the design principles, interface architecture and components is available in a modular and interlinked format so that users of the guide can quickly find answers to "how-to" questions.

This document is intended to be updated regularly as the Velocity UI evolves. New components and new functionality will be recorded here as they are approved by the team and through usability testing. The working assumption is that as similar design issues present themselves in future Velocity enhancements, their solutions will be reused from this guide.

Specific project information will be attached to the Style Guide as addendums. These addendums will detail the specific considerations which drive the instance of the interface found in the implementation of that project. Addendums should be restricted to articulating only the pertinent differences or notable specifics included in a project to reduce redundancy and assure the accuracy of the overall Style Guide. Where specific projects use existing interface components and conventions, guide users should be referred to the main Style Guide for answers.

The Velocity Style Guide is the single authoritative source of information regarding the Velocity UI. Developers should refer to the Style Guide whenever UI components are added to or changed in Velocity. Used as a tool to direct consistent User Interface development, this Style Guide will become a means toward unifying the interface design, improving the user experience and conveying Schwab's Brand.

If you have questions about the use or information contained in this style guide, please contact Constance Amador (65212) or Darin Rock (65629).

1.4 Organization of Style Guide

This first version of the Style Guide is divided into two main sections:

Building Velocity: the fundamentals. A brief overview of the primary components of the trading desktop, including an at-a-glance *Table of Velocity Components*.

Building Velocity: the components. An example of each component used in Velocity's current interface design with a definition of its attributes.

Building Velocity: the rules. An example of each component used in Velocity's current interface design with a definition of its attributes.

2 *Building Velocity: the fundamentals*

Velocity's interface design is constructed from three fundamental types of components: *containers*; *functional elements;* and *display elements*.

2.1 Building the base: the container

In its most fundamental aspect, Velocity's UI may be described as an integrated set of containers. A container is imply a window that contains other interface design components.

For the purposes of this style guide, a container is considered as a single component with varying attributes (or ways of describing what a container does). Most of a container's attributes will depend upon its contents—its specific combination of functional and display elements. But a few general parameters may help us define containers in terms of interface design:

Does this container need to be open for Velocity to run?

This attribute applies only to the base or program container. Every other container may be closed and Velocity will still be functional.

Is the container modal or non-modal?

A modal container is an active window that must be dismissed for the user to access any other part of Velocity. Containers with this attribute should only be used when a specific user operation is required for the program to continue.

The Log On window is an example is an example of a modal container.

Is the container layer-locked?

Some containers cannot be sent behind other containers: they always remain locked at the front of the user's screen. For example, Order Detail is a non-modal container that always appears at the front of the user's screen, even though the program container is accessible while Order Detail is open and interacts with the data displayed in Order Detail.

By contrast, the Order Entry window is a non-modal container whose layer is not locked: when the user clicks inside the program container, the Order Entry window is sent behind. When the user clicks inside the Order Entry screen again, it comes to the fore.

Is the container active or inactive?

The active/inactive distinction only applies to non-modal containers. (By definition, a modal container is always the active container.) The distinction is useful in defining a user scenario. We follow standard Windows usage: the active attribute may apply to any container at a given point, but only to one container at a time.

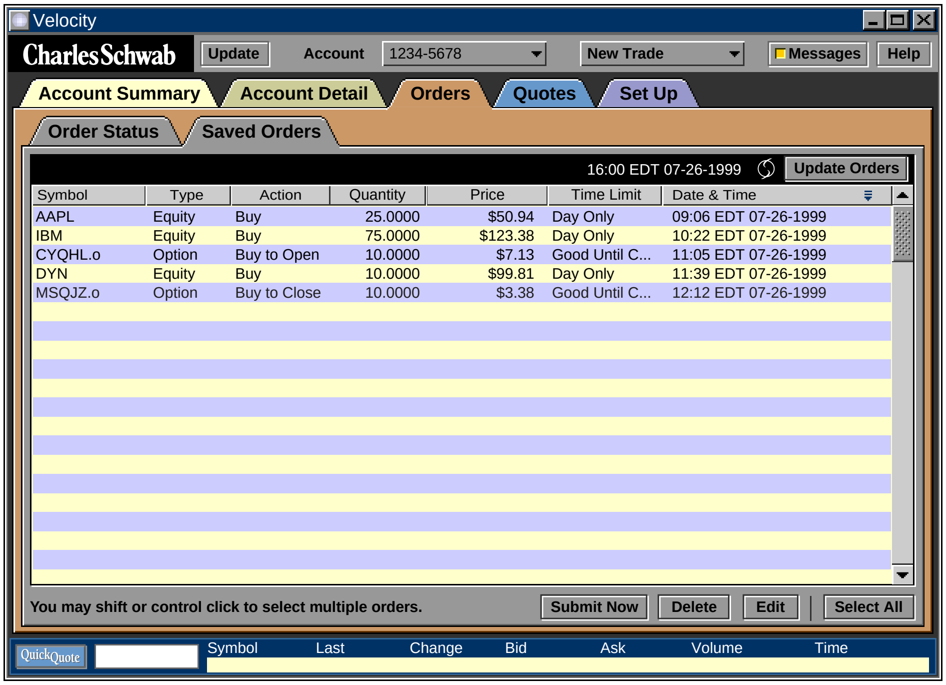
In the illustration above, the Order Status window is the active container.

What is the container's level of depth?

Some containers are nested in other containers, in which case the "level" attribute becomes relevant. For any set of nested containers, we define the outermost container as the base container (level = 0), and each subsequent set of nested containers in terms of its according to its degree of depth (level = 1,2,…).

For example, the illustration below has three separate levels of containers.

Level 0:
(the base container)
(nested) level 1
(nested) level 2



Symbol	Type	Action	Quantity	Price	Time Limit	Date & Time
AAPL	Equity	Buy	25.0000	$50.94	Day Only	09:06 EDT 07-26-1999
IBM	Equity	Buy	75.0000	$123.38	Day Only	10:22 EDT 07-26-1999
CYQHL.o	Option	Buy to Open	10.0000	$7.13	Good Until C...	11:05 EDT 07-26-1999
DYN	Equity	Buy	10.0000	$99.81	Day Only	11:39 EDT 07-26-1999
MSQJZ.o	Option	Buy to Close	10.0000	$3.38	Good Until C...	12:12 EDT 07-26-1999

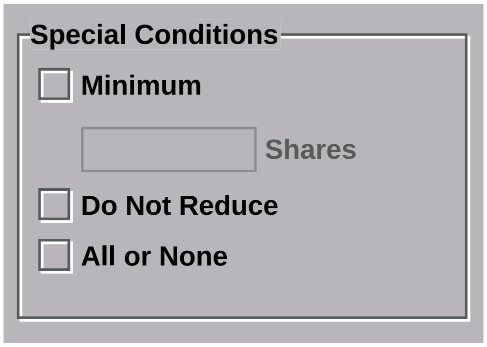
You may shift or control click to select multiple orders.

The base container contains functional elements, such as the NEW TRADE button in its top row, and the QUICK QUOTE button in its bottom row; the second level contains tab selects for primary program areas such as Account Summary, Account Detail, etc.; and the third level (the Order Status and Saved Status containers) is nested within the Account Summary container of the second level.

2.2 Building blocks: functional elements

A functional element is a graphical component that responds to a user action. The following pages list the functional elements used in the current version of Velocity.

2.2.1 check box



Special Conditions

□ Minimum

Shares

□ Do Not Reduce

□ All or None

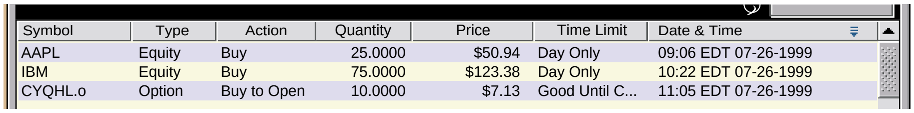
DESCRIPTION

A square box that is selected or cleared to turn on or off an option. More than one check box can be selected.

BEHAVIOR

A check appears when the square box is clicked by the users mouse: allows the user to indicate a decision or to make one or more selections from a list. (cp. radio button).

2.2.2 column sort button



Symbol	Type	Action	Quantity	Price	Time Limit	Date & Time
AAPL	Equity	Buy	25.0000	$50.94	Day Only	09:06 EDT 07-26-1999
IBM	Equity	Buy	75.0000	$123.38	Day Only	10:22 EDT 07-26-1999
CYQHL.o	Option	Buy to Open	10.0000	$7.13	Good Until C...	11:05 EDT 07-26-1999

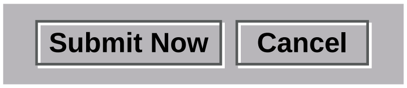
DESCRIPTION

A beveled rectangle set at the top of a grid column, containing a text heading and a down arrow.

BEHAVIOR

When enabled, allows the user to sort the contents of a specific column. In the illustration shown above, clicking on the Date & Time button will sort the quote symbols by the date and time traded. ??

2.2.3 command button



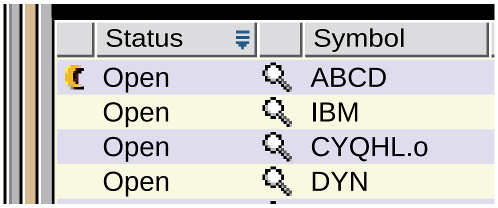
DESCRIPTION

A beveled (usually) rectangle containing text indicating its function.

BEHAVIOR

Executes the action designated by its text when clicked on by the user.

2.2.4 icon button



	Status		Symbol
☾	Open	🔍	ABCD
	Open	🔍	IBM
	Open	🔍	CYQHL.o
	Open	🔍	DYN

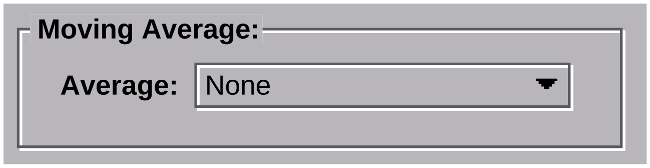
DESCRIPTION

A button in the form of a graphic symbol.

BEHAVIOR

Executes the action designated by its symbol when clicked on by the user. In the illustration shown above, there are two icons, but only the magnifying glass symbol is a button—which launches the Quote Detail container when clicked on the the user. The crescent moon symbol is the After Hours icon; it is a display element (see _____). Nothing happens if the user clicks on it. Velocity helps users distinguish functional from display elements with its Tool Tip feature (see _____).

2.2.5 pulldown button



Average: None

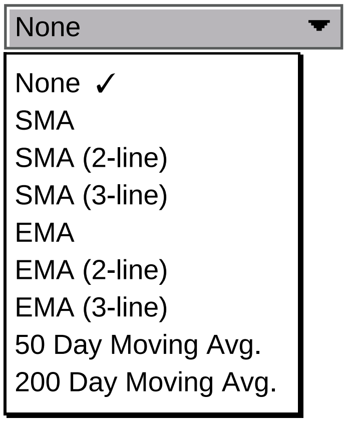
DESCRIPTION

A beveled rectangle containing text and a down arrow.

BEHAVIOR

The down arrow next to the text indicates that this button will reveal a pulldown menu when user moves the mouse over the button. The text appearing in the button is the currently selected item from the menu (see next component).

2.2.6 pulldown menu



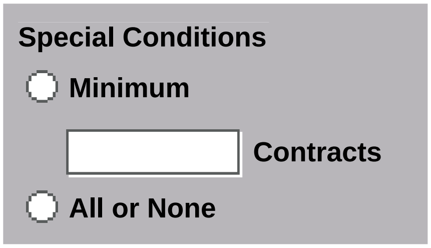
DESCRIPTION

A list display that appears when the user moves the mouse over a pulldown button (see preceding item).

BEHAVIOR

Allows the user to select any (enabled) item from the list. The checked item is the currently selected item.

2.2.7 radio button



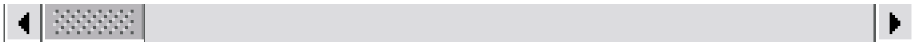
DESCRIPTION

A round button that is selected or cleared to turn on or off an option.

BEHAVIOR

Allows the user to make one choice from among several options listed in a dialog box. In the dialog shown above, the user may select either “Minimum” or “All or None.”

2.2.8 scroll bar



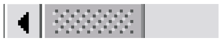
DESCRIPTION

A rectangular strip, usually located at the left and or bottom border of a container, containing two arrow buttons and a sliding strip button (see next component).

BEHAVIOR

Standard Windows usage: allows the user to scroll back and forth or up and down within a window that has a wider view than what can be displayed within the container.

2.2.9 scroll bar buttons



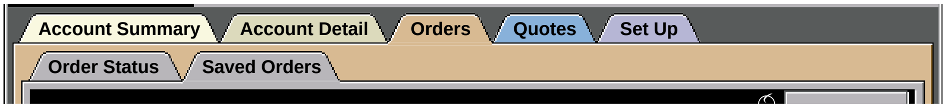
DESCRIPTION

Within a scroll bar (see preceding component), one of two types of button: (1) a square button containing an arrow; or (2) a stippled slider button.

BEHAVIOR

Standard Windows usage: when the user holds the mouse down over the square button, the window scrolls in the direction of the arrow . When the user holds the mouse down over the slider button and drags, the window scrolls in the direction of the drag.

2.2.10 tab select



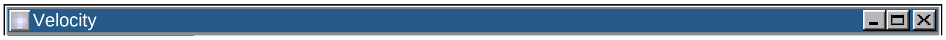
DESCRIPTION

A curved rectilinear rhombus button shaped to look like the tab of an office folder, displaying the title of a nested container.

BEHAVIOR

When the user clicks on a tab select, the container indicated by the text title becomes active (moves to the fore). Used for nested containers. In the illustration shown above two levels of multiple nested containers are accessible to the user.

2.2.11 title bar buttons



DESCRIPTION

Beveled rectangular buttons set at th eleft end of a title bar, typically three in number: contains the standard Windows symbols for “minimize,” “maximize,” and “close.”

BEHAVIOR

Standard Windows usage: when the user clicks:

− container window is minimized.

□ container window is maximized.

☒ container window is closed.

Only the “close” button is required by all containers; the first two are optional.

2.3 Building blocks: display elements

A display element is a graphical component that visually organizes the user interface. Because the current version of Velocity does not include training or support material to guide the user through its functions, the UI is designed to be as obvious as possible.

to come

3 *Building Velocity: the rules*

3.1 Current constraints

Building user interfaces in Velocity 2.x to accommodate new or revised functionality is an exercise combing business requirements, system restraints, and the guidelines and components described in this document. A series of questions and answers, following a general order, should result in choices from this guide for containers and components which ultimately form the completed user interface. When there are no existing solutions within the existing user interface, new ones will be invented.

Efficiency is realized by using as much of the existing organization and reusing as many components as possible. Usability is improved by using understood conventions and maintaining consistency.

3.2 Choosing a container

Container attributes are discussed in Chapter 2 (“Building the base: the container” on page 6).Velocity features several “key” functional areas already delineated by containers. These are:

- Account Overview
- Account Detail
- Orders
- Quotes
- Set-Up
- Trade Entry

New or additional functionality should first be evaluated to determine which of these existing spaces would be appropriate. Use the following guidelines:

Account Overview. Provides a limited summary view of positions, balances and order status. Because it is an overview, detailed information or additional navigations are avoided in the Account Overview container.

Account Detail. Provides extended balances, positions and transaction information. Account status or account maintenance features are good candidates for this space.

Orders. Provides information on currently active and saved trades. Ability to change, modify, cancel or review status on orders fits this area functionality.

Quotes. Contains most quoting and, to some degree, monitoring and research tools. The Quotes space should be considered for additional research and tracking tools.

Set up. A variety of system and online account configuration tools are provided, or should be implemented in the Set-Up space.

Trade entry. Dedicated to the variety of appropriate orders.