



# VA Smalltalk Code Completion

Seth Berman  
Instantiations, Inc.

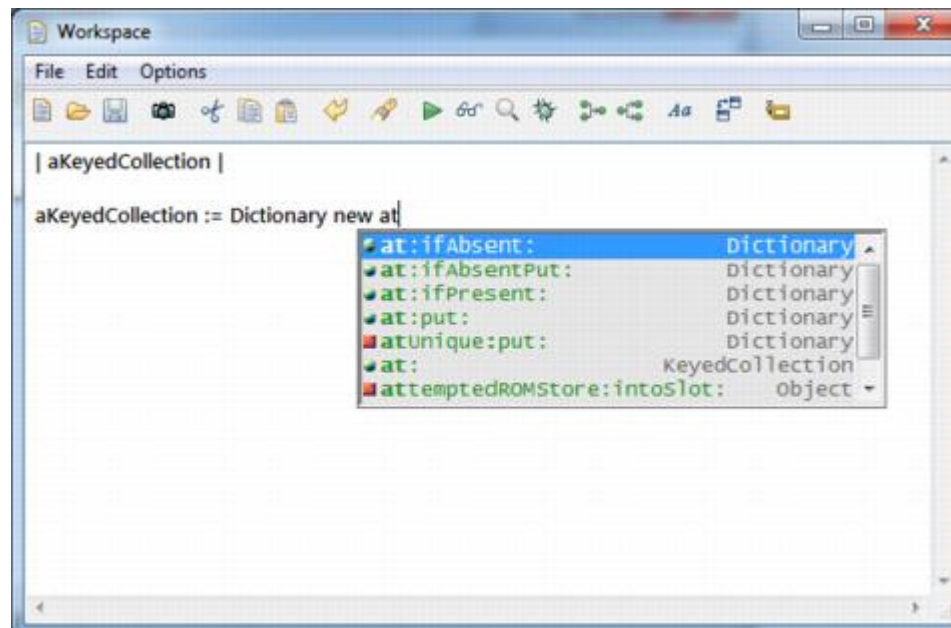
---

# Who Am I?

- Began using Smalltalk in 2001
  - 8 years total Smalltalk experience
  - JWARS Simulation Developer
- 3.5 years outside the Smalltalk arena
  - Providing Government Solutions
  - Data-Ingestion (ETL)
  - Link-Analysis
  - Visualization Tools
  - Grid Computing
- Joined Instantiations in May 2011

# Code Completion Introduction

- **Problem:** Our ability to recall names is limited
- **Solution:** Use the computer's calculating capabilities to predict names as they are being entered in the text editor



---

# Code Completion Release History

- V8.5 (August 2011)
  - First Release
  - Requirements derived from community discussion
  - Additional requirements based on IDE experiences
  - Lots of Research
- V8.5.1 (Upcoming Release)
  - Architectural improvements
  - Many new features
  - Emphasis on exposing configuration options

---

# Code Completion - V8.5.1

## What's New?

- Redesigned Configuration UI
- Theme & Match Highlighting
- CamelCase Matching
- Live-Filtering
- Block Argument Detection
- Prereq Chain Visibility
- Drag-N-Drop Support
- New Completion Types

---

# Code Completion Design

- Build general content assist framework
  - Minimal Dependencies
  - Pluggable
  - Extensible
- Phase 1 – Extend framework to provide the first code assist feature...code completion
- Future Phases
  - Additional code assist features
  - Unify existing code assist features

---

# Code Completion - Actors

- **Code Assistant:** Mediates the interactions between the various actors
- **Context:** Provides additional information about the environment that the Text Widget lives in
- **Provider:** Provides suggestions in response to a query
- **Suggestion:** Actionable item that can be rendered in the Popup

---

# Code Completion – Actors

## continued...

- **Text Widget:** Bridge abstraction that wraps a text widget implementation
- **Popup:** Bridge abstraction that wraps a container widget implementation
- **Session:** Tracks the state of computation during a content assist workflow



---

# Code Completion - Contexts

- Code completion is available in 3 contexts
  - Browsers
  - Debuggers
  - Workspaces
- Context information is used to help resolve types
- Contexts help define what classes/methods are visible

---

# Code Completion - Suggestions

- Variables
  - Local
  - Instance/Class/Class Instance
  - Pseudo
  - Pool
  - Global
- Classes
- Methods
- Symbols/Atoms
- Pool Dictionaries
- Character Literals

---

# Code Completion - Session

- Sessions perform 4 basic functions
  - Queries
  - Filters
  - Restoring State
  - Caching
- When code completion is activated, a session is opened
- When code completion ends, the session is closed

---

# Code Completion – Work Flow

- On activation, have @150ms to...
  1. Prepare the text from the cursor to the top of source
  2. Create parse tree
  3. Identify context
  4. Perform static analysis
  5. Identify candidate completions (match algorithms)
  6. Draw suggestions in the popup

---

# Code Completion Demo

# DEMO TIME