



*TOMORROW  
starts here.*

Cisco *live!*



# The Essential Quick Start Guide To Adopting Finesse As The Agent Desktop Interface

BRKCCT-2559

Paul Tindall

EMEAR Customer Collaboration Technical Consulting

@tindallpaul, [ptindall@cisco.com](mailto:ptindall@cisco.com)

Cisco *live!*

This session is a practical getting started guide to building the agent desktop the way you want it using Finesse. In 90 minutes you will gain the essential knowledge to start working hands-on with the Finesse desktop, configuring and customising it to meet business requirements.

All examples and desktop content presented here is authentic and has been created using a freshly installed working platform as the starting point.

# Agenda

- Introduction
- Configuration
- Adding custom gadgets
- Embedding web content
- Workflow action event handling
- Call control and state change events
- Alternative approaches
- Getting started

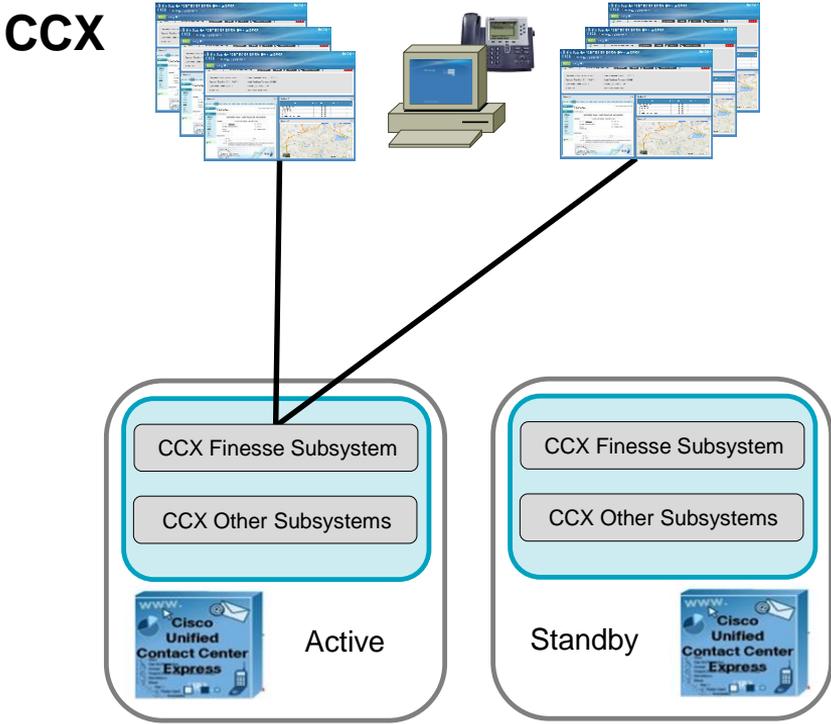
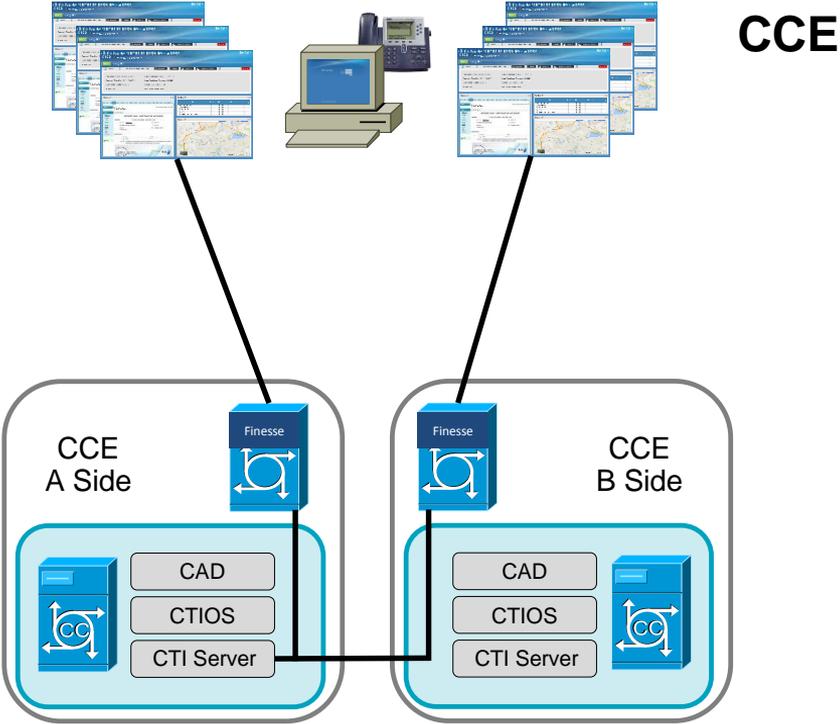


# Introduction

# What is Finesse?

- Desktop interface to CCE, PCCE and CCX
- Has web-based architecture, meaning in simple terms ...
  - Standard web browser (Microsoft IE / Firefox) as the desktop user interface
  - Communicating over HTTP to web server(s)
  - Behind that ...
    - Application server components
    - Internal database of configuration information such as reason codes, phone books
    - Backend interface to CCX / CCE CTI servers
  - User connects via web URL, signs on, served with dynamically generated web content

# Finesse Very High-Level



# At The User-Side

- Web page with modular structure, comprising ...
  - Call control / status component
  - Tabbed layout
  - Tabs optionally divided into columns
  - Columns populated by one or more Finesse modules (Gadgets)

The screenshot shows a Cisco Finesse agent console for Agent Ann Dronicus (Extension 4410). The interface is modular and includes the following components:

- Header:** Agent name, extension, and a 'Sign Out' button.
- Navigation:** Tabs for Home, My Statistics, Manage Call, Charts, Layout, Map, Breakouts, and Session.
- Call Information:** Call duration (00:05), Attendee ID (106717), and Call ID (07690024280). Buttons for Keypad, Hold, and Consult are visible.
- Agent Profile:** Name (Isaac Mason), Company (Boyle and Lancet), Mobile (07918398436), Attendee ID (106717), Breakout Type (BRKCCT), and Session ID (2559). A 'Wrap-Up Reason' dropdown and 'Apply' button are also present.
- Security/Travel/Support:** A row of buttons for Security, Travel, and Support.
- Breakouts:** A section titled 'Cisco Live Milan Contact Centre Tech Breakouts' with a form for Name, Mobile, Email, Company, Job Title, and Attendee ID.
- Custom Charts - Agent State:** A bar chart titled 'Agent State Summary' showing the count of agents in various states: Ready (20), Not Ready (1), Talking (5), Hold (1), After Call Work (1), Work Not Ready (1), Work Ready (1), and Reserved (1).
- Map - Cisco Live Milan:** A map showing the location of 'MiCo Milano Congressi' with a red pin and a 'View on Google Maps' link.

# Finesse Gadgets – What Actually Are They?

- Widget ...
  - Reusable chunk of standard web code – HTML, JavaScript, CSS
  - Would execute and render in almost any web page
- Gadget ...
  - Logically the same
  - It's still a chunk of web code, but
  - Built to fit specific type of container
  - Wrapped in specific XML format
  - OpenSocial specification in the case of Finesse gadgets

The screenshot displays the Cisco Finesse agent console interface. At the top, it shows the agent's name 'Agent Ann Dronicus (andronicus) - Extension 4410' and a 'Sign Out' button. Below this is a navigation bar with options like 'Home', 'My Statistics', 'Manage Call', 'Charts', 'Layout', 'Map', 'Breakouts', and 'Session'. The main area shows call details: '00:05 | Attendee ID: 106717 | 07690024280'. A 'Wrap-Up Reason' dropdown and 'Apply' button are visible. Below the call details are tabs for 'Security', 'Travel', and 'Support'. Three gadgets are highlighted with black circles: 1. 'Cisco Live Milan Contact Centre Tech Breakouts' form with fields for Name (Isaac Mason), Mobile (07918398436), Email (isaacmason@boylelan), Company (Boyle and Lancel), Job Title (Solution Designer), and Attendee ID (106717). 2. 'Custom Charts - Agent State' bar chart titled 'Agent State Summary' showing counts for Ready, Not Ready, Talking, Hold, After Call Work, Work Not Ready, Work Ready, and Reserved. 3. 'Map - Cisco Live Milan' showing a map with a red pin at 'MiCo Milano Congressi' and a 'View on Google Maps' link.

# Why is Finesse Important?

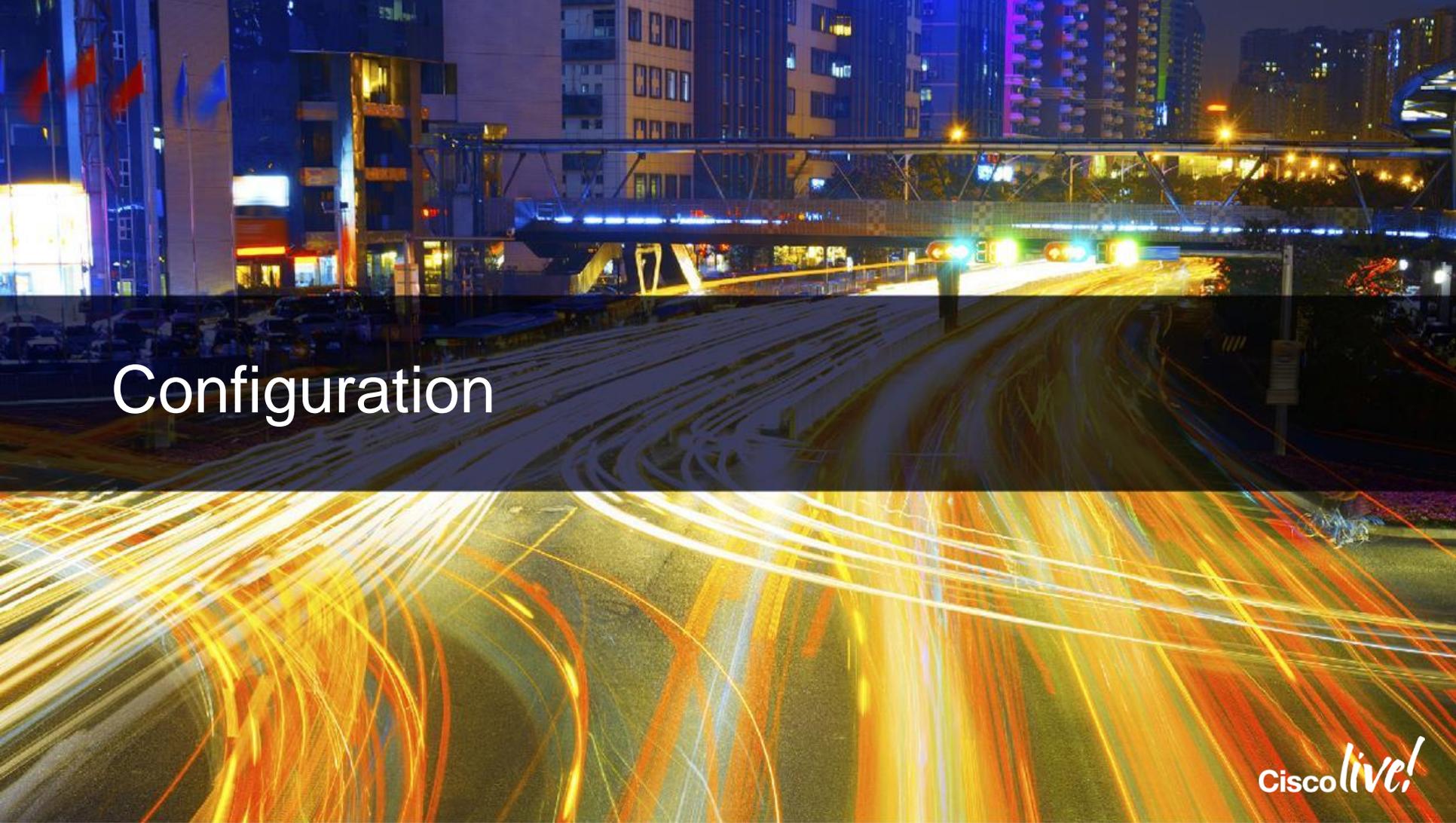
- Replaces CAD and CTIOS
  - Use for new deployments plus very many existing desktops to migrate
- Applies to both CCX and CCE
- No more thick client desktop applications
  - No more install/upgrade pain across a huge number of desktops
- Opportunity to create customised business & user-role relevant desktop content
- Modular content
  - Build and test modules independently
  - Make available by adding to the content layout definition
  - Team specific layouts

# CAD/CTI-OS End Of Life Summary

- EOL process started Jan 2015 for both CCE & CCX
- CCX
  - 10.6 will be the last release of CAD for CCX
- CCE
  - 11.0 will include CTI-OS and CAD for upgrades only
  - Finesse is the only desktop option for new CCE 11.0 customers
- CTI-OS will only continue to be supported for ICM

# Getting Started – Approach

- Install/add Finesse server to existing CCE, enable as option within CCX
  - CCE: Additional settings for CTI Server and AW DB
  - CCX: explicitly enable using command line CLI – **utils uccx finesse activate**
- Use it in parallel with other desktop flavours
  - CCX: 10.6 release planned to support concurrent use of CAD and Finesse (end Feb 2015)
- First steps
  - Use the out-the-box desktop
  - Configuration using Finesse Administration
    - Screen presentation / layouts
    - Reason codes
    - Phonebooks
    - Automated workflow triggers and actions
- Then build/add content modules – Cisco-provided, third-party or custom built



# Configuration

# Configuration

## Finesse Administration

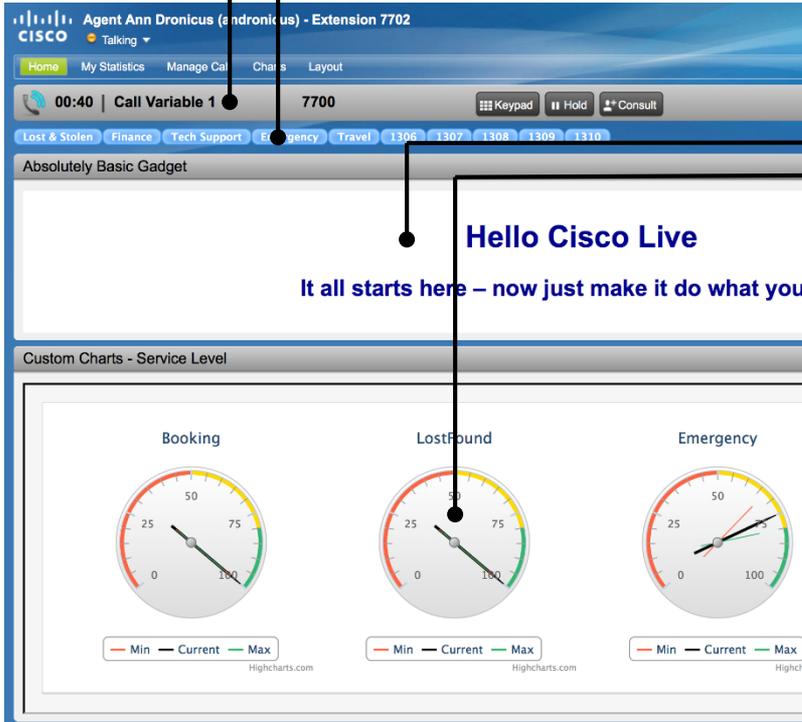
- CCX – Integrated with App Admin
- Desktop gadget layout
- Call variable presentation
- Phonebooks
- Reason codes
- Not-Ready
- Sign-Out
- Wrap-Up
- Event-driven workflows
- Finesse API includes configuration commands

The screenshot displays the Cisco Unified CCX Administration web interface. At the top, there is a navigation menu with options like 'Disaster Recovery System', 'Cisco Unified CCX Administration', 'Cisco Desktop Administrator', 'Cisco Finesse Administration', 'Cisco Unified Serviceability', 'Cisco Unified CCX Serviceability', and 'Cisco Unified OS Administration'. The main content area is titled 'Cisco Finesse Administration' and includes a 'Manage Team Resources' section. This section contains a 'List of Teams' table with columns for Name and ID. Below the table is a 'Refresh' button. Further down, there is a 'Resources for Cisco Live Milan' section with tabs for 'Desktop Layout', 'Phone Books', 'Reason Codes (Not Ready)', 'Reason Codes (Sign Out)', 'Wrap-Up Reasons', and 'Workflows'. The 'Desktop Layout Configuration' section is active, showing a checkbox for 'Override System Default' and a 'Desktop Layout XML' field containing XML code for configuring the desktop layout.

Name	ID
Cisco Live Milan	6
Consulting	4
Default	1
Helpdesk	2
Tech Support	3

```
<finesseLayout xmlns="http://www.cisco.com/vtg/finesse">
  <layout>
    <role>Agent</role>
    <page>
      <gadget>/desktop/gadgets/CallControl.jsp</gadget>
      <gadget>http://10.52.202.40:8080/fs/buttons/CustomTransferButtonArray.xml</gadget>
    </page>
    <tabs>
      <tab>
        <id>home</id>
        <label>finesse.container.tabs.agent.homeLabel</label>
      </tab>
      <gadget>http://10.52.202.40:8080/fs/starthere/AbsolutelyBasicGadget.xml</gadget>
      <gadget>http://10.52.202.40:8080/fs/charts/CustomChartsService.xml</gadget>
    </tabs>
  </layout>
</finesseLayout>
```

# Desktop Layout

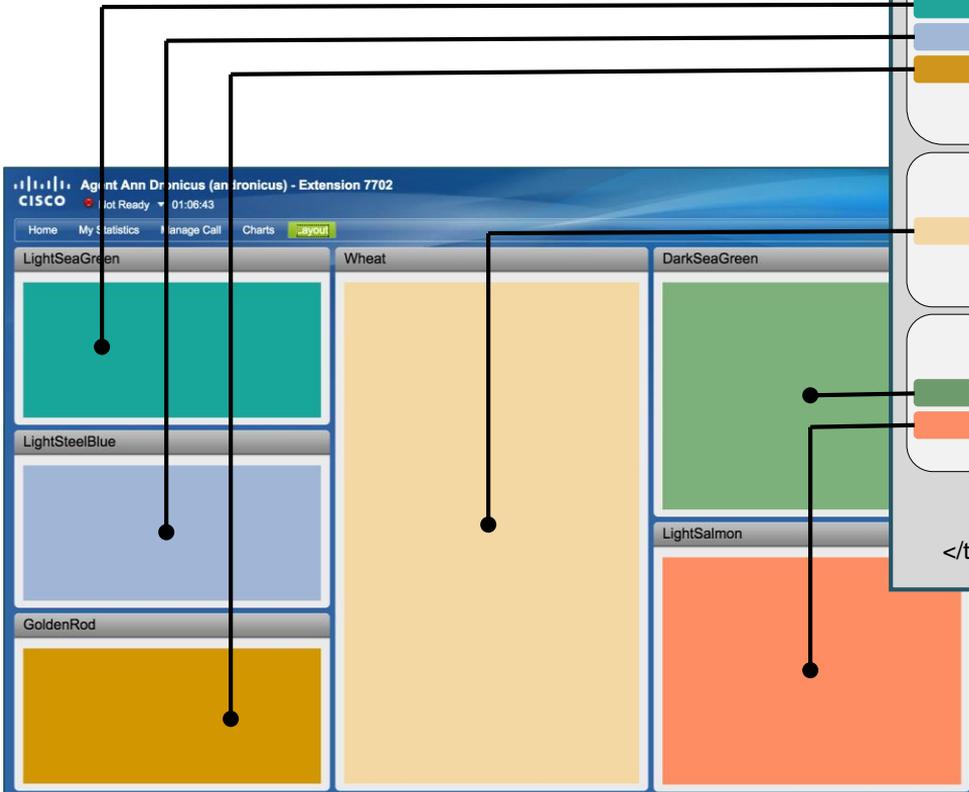


```
<finesseLayout xmlns="http://www.cisco.com/vtg/finesse">
  <layout>
    <role>Agent</role>
    <page>
      <gadget>/desktop/gadgets/CallControl.jsp</gadget>
      <gadget>http://10.52.202.40:8080/fs/TransferButtonArray.xml</gadget>
    </page>
    <tabs>
      <tab>
        <id>home</id>
        <label>finesse.container.tabs.agent.homeLabel</label>
        <gadgets>
          <gadget>http://10.52.202.40:8080/fs/AbsolutelyBasicGadget.xml</gadget>
          <gadget>http://10.52.202.40:8080/fs/CustomChartsService.xml</gadget>
        </gadgets>
      </tab>
      <tab>
        <id>myStatistics</id>
        <label>finesse.container.tabs.agent.myStatisticsLabel</label>
        ...
      </tab>
    </tabs>
  </layout>
</finesseLayout>
```

Cisco live!

# Desktop Gadget Layout

## Multiple Columns



```
<tab>
  <id>layout</id>
  <label>Layout</label>
  <columns>
    <column>
      <gadgets>
        <gadget>http://10.52.202.40:8080/fs/col/LightSeaGreen.xml</gadget>
        <gadget>http://10.52.202.40:8080/fs/col/LightSteelBlue.xml</gadget>
        <gadget>http://10.52.202.40:8080/fs/col/GoldenRod.xml</gadget>
      </gadgets>
    </column>
    <column>
      <gadgets>
        <gadget>http://10.52.202.40:8080/fs/col/Wheat.xml</gadget>
      </gadgets>
    </column>
    <column>
      <gadgets>
        <gadget>http://10.52.202.40:8080/fs/col/DarkSeaGreen.xml</gadget>
        <gadget>http://10.52.202.40:8080/fs/col/LightSalmon.xml</gadget>
      </gadgets>
    </column>
  </columns>
</tab>
```

# Reason Codes

Maximum number of codes:

(Non-global codes are assigned to specific teams)

	Global	Non-Global
Not Ready	100	100
Sign Out	100	100
Wrap Up	100	100

Cisco Finesse Administration

Call Variables Layout Desktop Layout Phone Books **Reasons** Team Resources Workflows

Manage Reason Codes (Not Ready)

Reason Label	Reason Code	Global?
Ad Hoc Meeting	100	Yes
Analysis	101	No
Follow-up Actions	102	Yes
Direct Call		
Back Office		
Researching		

New Edit Del

Manage Reason Codes (Sign Out)

Reason Label	Reason Code	Global?
End of Day	100	Yes
Lunch Break	101	Yes
Briefing	102	Yes
Break	103	Yes
Admin		
Training		
Lab Investigation		
Meet The Engineer		

New Edit Del

Manage Wrap-Up Reasons

Wrap-Up Reason Label	Global?
Customer Pending	Yes
Following-Up	Yes
Case Closed	Yes
Escalated	No
Cancelled	Yes
Resolved	Yes
Handed Off	Yes
Complaint	Yes
Developer Query	No

New Edit Delete Refresh



# Reason Codes

Cisco Finesse Administration

Call Variables Layout Desktop Layout Phone Books **Reasons** Team Resources

### Manage Reason Codes (Not Ready)

Reason Label	Reason Code
Ad Hoc Meeting	100
Analysis	101
Follow-up Actions	102
Direct Call	
Back Office	
Researching	

New Edit Delete

### Manage Reason Codes (Sign Out)

Reason Label
End of Day
Lunch Break
Briefing
Break
Admin
Training
Lab Investigation
Meet The Engineer

New Edit Delete

### Manage Wrap-Up Reasons

Wrap-Up Reason Label
Customer Pending
Following-Up
Case Closed
Escalated
Cancelled
Resolved
Handed Off
Complaint
Developer Query

New Edit Delete Refresh

Agent Ann Dronicus (andronicus) - Extension 7702

Talking

- Ready
- Not Ready - Ad Hoc Meeting
- Not Ready - Back Office
- Not Ready - Direct Call
- Not Ready - End of Session
- Not Ready - Follow-up Actions
- Not Ready - Researching

07898117106

Sign Out

- Admin
- Break
- Briefing
- End of Day
- Lunch Break
- Meet The Engineer
- Training

Sign Out

- Customer Pending
- Following-Up
- Handed Off
- Resolved

Apply

Reason code pull-down lists resulting from combination of team and global settings

# Team Specific vs. Global Settings

The screenshot displays the Cisco Finesse Administration interface. At the top, the navigation bar includes 'Call Variables Layout', 'Desktop Layout', 'Phone Books', 'Reasons', 'Team Resources', and 'Workflows'. The 'Team Resources' tab is active, showing a 'List of Teams' table with columns 'Name' and 'ID'. Below this is a 'Resources for Consulting' section with tabs for 'Desktop Layout', 'Phone Books', 'Reason Codes (Not Ready)', 'Reason Codes (Sign Out)', 'Wrap-Up Reasons', and 'Workflows'. The 'Reason Codes (Not Ready)' tab is selected, showing a table with columns 'Label' and 'Code'. At the bottom, there are 'Save' and 'Revert' buttons.

Name	ID
Cisco Live Milan	6
Consulting	4
Default	1
Helpdesk	2
Tech Support	3

Label	Code
Analysis	101
Back Office	104
Researching	105

## Tabs for each category of global settings

- Call variable presentation
- Desktop gadget layout
- Phone books
- Reason codes
- Team overrides
- Event-driven workflows

## Overriding global settings

### Tabs for each category of team specific settings

- Desktop gadget layout
- Phone books
- Reason codes  
(Not Ready, Sign Out, Wrap-Up)
- Event-driven workflows

Cisco *live!*

# Team Specific vs. Global Settings

The screenshot shows the Cisco Finesse Administration interface. The top navigation bar includes 'Call Variables Layout', 'Desktop Layout', 'Phone Books', 'Reasons', 'Team Resources', and 'Workflows'. The 'Team Resources' tab is selected, showing a 'List of Teams' table with columns 'Name' and 'ID'. The 'Consulting' team is highlighted. Below this is a 'Resources for Consulting' section with tabs for 'Desktop Layout', 'Phone Books', and 'Reason Codes (Not Ready)'. The 'Desktop Layout Configuration' section shows an XML editor for 'Desktop Layout XML' with a 'Save' button. A callout box points to the 'Team Resources' tab, listing global settings categories. Another callout box points to the 'Reason Codes (Not Ready)' tab, listing team-specific settings categories.

Name	ID
Cisco Live Milan	6
Consulting	4
Default	1
Helpdesk	2
Tech Support	3

Label	Code
Analysis	101
Back Office	104
Researching	105

Tabs for each category of **global settings**

- Call variable presentation
- Desktop gadget layout
- Phone books
- Reason codes
- Team overrides
- Event-driven workflows

**Overriding global settings**

Tabs for each category of **team specific settings**

- Desktop gadget layout
- Phone books
- Reason codes (Not Ready, Sign Out, Wrap-Up)
- Event-driven workflows

Cisco live!

# Phone Books

## Manage Phone Books

### List of Phone Books

Name ▲	Assign To
Admin Department	All Users
Cisco Live	Teams
Tech Teams	All Users
UK Office	All Users

[New](#) [Edit](#) [Delete](#) [Refresh](#)

### List of Contacts for UK Office

Last Name	First Name ▲	Number	Note
Fletcher	Alice	07789021349	
Cook	Ann	04647615512	
Slater	Bob	04283886140	
Tyler	Carol	04143940257	
Smith	Janet	07883344556	
Carpenter	Jim	04203506413	
Potter	John	08306024613	
Cordwainer	Joseph	01794432540	
Miller	Michael	09180398772	
Palmer	Phil	09730022002	

[New](#) [Edit](#) [Delete](#) [Refresh](#)

[Import](#) [Export](#)



# Phone Books

Cisco Finesse Administration Sign Out

Call Variables Layout Desktop Layout **Phone Books** Reasons Team Resources Workflows

### Manage Phone Books

#### List of Phone Books

Name	Assign To
Admin Department	All Users
Cisco Live	Teams
Tech Teams	All Users
UK Office	All Users

New Edit Delete Refresh

#### List of Contacts for UK Office

Last Name	First Name	Number	Note
Fletcher	Alice	07789021349	
Cook	Ann	04647615512	
Slater	Bob	04283886140	
Tyler	Carol	04143940257	
Smith	Janet	07883344556	
Carpenter	Jim	04203506413	
Potter	John	08306024613	
Cordwainer	Joseph	01794432540	
Miller	Michael	09180398772	
Dever	Paul	09720022002	

New Edit Delete Refresh Import Export

# Phone Books

Cisco Finesse Administration Sign Out

Call Variables Layout Desktop Layout **Phone Books** Reasons Team Resources Workflows

### Manage Phone Books

#### List of Phone Books

Name	Assign To
Admin Department	All Users
Cisco Live	Teams
Tech Teams	All Users
UK Office	All Users

**New** **Edit** **Delete** **Refresh**

#### List of Contacts for UK Office

Last Name	First Name	Number	Note
Fletcher	Alice	077890	
Cook	Ann	046476	
Slater	Bob	042838	
Tyler	Carol	041439	
Smith	Janet	078833	
Carpenter	Jim	042035	
Potter	John	083060	
Cordwainer	Joseph	017944	
Miller	Michael	091803	
Baker	Phil	097390	
Sawyer	Sarah	077885	Fictitious person for UK phone boo
Butcher	Ted	076198	
Taylor	Thomas	019771	

**New** **Edit** **Delete** **Refresh** **Import** **Export**

#### List of Contacts for UK Office

Last Name	First Name	Number	Note
Smith	Janet	07883344556	
Carpenter	Jim	04203506413	
Potter	John	08306024613	
Cordwainer	Joseph	01794432540	
Miller	Michael	09180398772	
Baker	Phil	09739033082	
Sawyer	Sarah	07788534212	Fictitious person for UK phone boo
Butcher	Ted	07619806240	
Taylor	Thomas	01977167129	

**Import Contacts**

All existing Contacts in the UK Office Phone Book will be replaced with the file you choose:

Contacts File  **Browse...**

**OK** **Cancel**

# Phone Books

Cisco Finesse Administration Sign Out

Call Variables Layout Desktop Layout **Phone Books** Reasons Team Resources Workflows

### Manage Phone Books

#### List of Phone Books

Name	Assign To
Admin Department	All Users
Cisco Live	Teams
Tech Teams	All Users
UK Office	All Users

#### List of Contacts for UK Office

Last Name	First Name	Number
Fletcher	Alice	077890
Cook	Ann	046476
Slater	Bob	042838
Tyler	Carol	041439
Smith	Janet	078833
Carpenter	Jim	042035
Potter	John	083060
Cordwainer	Joseph	017944
Miller	Michael	091803
Baker	Phil	097390

#### List of Contacts for UK Office

Last Name	First Name	Number	Note
Smith	Janet	07883344556	
Carpenter	Jim	04203506413	
Potter	John	08306024613	
Cordwainer	Joseph	01794432540	
Miller	Michael	09180398772	
Baker	Phil	09739033082	
Sawyer	Sarah	07788534212	Fictitious person for UK phone boo
Butcher	Ted	07619806240	
Taylor	Thomas	01977167129	

**Import Contacts**

All existing Contacts in the UK Office Phone Book will be replaced with the file you choose:

Contacts File

```
UC-Contacts.csv
"First Name","Last Name","Phone Number","Notes"
"Bob","Slater","04283886140",""
"Thomas","Taylor","01977167129",""
"Phil","Baker","09739033082",""
"Michael","Miller","09180398772",""
"Ann","Cook","04647615512",""
"Ted","Butcher","07619806240",""
"Carol","Tyler","04143940257",""
"Joseph","Cordwaine","01794432540",""
"Sarah","Sawyer","07788534212","Fictitious person for UK phone book"
"Alice","Fletcher","07789021349",""
"Janet","Smith","07883344556",""
"John","Potter","08306024613",""
"Jim","Carpenter","04203506413",""
```

# Phone Books

Cisco Finesse Administration Sign Out

Call Variables Layout Desktop Layout **Phone Books** Reasons Team Resources Workflows

### Manage Phone Books

#### List of Phone Books

Name	Assign To
Admin Department	All Users
Cisco Live	Teams
Tech Teams	All Users
UK Office	All Users

#### List of Contacts for UK Office

Last Name	First Name	Number
Fletcher	Alice	077890
Cook	Ann	046476
Slater	Bob	042838
Tyler	Carol	041439
Smith	Janet	078833
Carpenter	Jim	042035
Potter	John	083060
Cordwainer	Joseph	017944
Miller	Michael	091803
Baker		0072022002

#### List of Contacts for UK Office

Last Name	First Name	Number
Smith		
Carpenter		
Potter		
Cordwainer		
Miller		
Baker		
Sawyer		
Butcher		
Taylor		

#### New Contact

First Name  Number

Last Name  Note

#### Import Contacts

All existing Contacts in the UK Office Phone Book will be replaced with the file you choose:

Contacts File

```
"First Name","Last Name","Phone Number","Notes"
"Bob","Slater","04283886140",""
"Thomas","Taylor","01977167129",""
"Phil","Baker","09739033082",""
"Michael","Miller","09180398772",""
"Ann","Cook","04647615512",""
"Ted","Butcher","07619806240",""
"Carol","Tyler","04143940257",""
"Joseph","Cordwainer","01794432540",""
"Sarah","Sawyer","07788534212","Fictitious person for UK phone book"
"Alice","Fletcher","07789021349",""
"Janet","Smith","07883344556",""
"John","Potter","08306024613",""
"Jim","Carpenter","04203506413",""
```

# Phone Books

Cisco Finesse Administration

Call Variables Layout Desktop Layout **Phone Books** Reasons Team Resources Workflows

### Manage Phone Books

#### List of Phone Books

Name	Assign To
Admin Department	All Users
Cisco Live	Teams
Tech Teams	All Users
UK Office	All Users

New Edit Delete Refresh

#### List of Contacts for UK Office

Last Name	First Name	Number
Fletcher	Alice	077890
Cook	Ann	046476
Slater	Bob	042838
Tyler	Carol	041439
Smith	Janet	078833
Carpenter	Jim	042035
Potter	John	083060
Cordwainer	Joseph	017944
Miller	Michael	091803
Baker		00720022002

New Edit Delete Refresh

Import Export

## Phone Book Limits

(Total contacts planned to be increased in 11.0)

Global Phone Books	10
Team Phone Books	50
<b>Total Configured Contacts</b>	<b>1500</b>

#### List of Contacts for UK Office

Last Name	First Name	Number	Note
Fletcher	Alice	07789021349	
Cook	Ann	04647615512	
Slater	Bob	04283886140	
Tyler	Carol	04143940257	
Smith	Janet	07883344556	
Carpenter	Jim	04203506413	
Potter	John	08306024613	
Cordwainer	Joseph	01794432540	
Miller	Michael	09180398772	
Baker		00720022002	

#### New Contact

First Name: Sarah Number: 07788534212  
 Last Name: Sawyer Note: Fictitious person for UK phone book

Save Cancel

#### List of Contacts for UK Office

Last Name	First Name	Number
Smith		
Carpenter		
Potter		
Cordwainer		
Miller		
Baker		
Sawyer		
Butcher		
Taylor		

#### Import Contacts

All existing Contacts in the UK Office Phone Book will be replaced with the file you choose:

Contacts File:  Browse...

OK Cancel

```

"First Name","Last Name","Phone Number","Notes"
"Bob","Slater","04283886140",""
"Thomas","Taylor","01977167129",""
"Phil","Baker","09739033082",""
"Michael","Miller","09180398772",""
"Ann","Cook","04647615512",""
"Ted","Butcher","07619806240",""
"Carol","Tyler","04143940257",""
"Joseph","Cordwainer","01794432540",""
"Sarah","Sawyer","07788534212","Fictitious person for UK phone book"
"Alice","Fletcher","07789021349",""
"Janet","Smith","07883344556",""
"John","Potter","08306024613",""
"Jim","Carpenter","04203506413",""
  
```

# Call Variable Data Presentation

- Can select one variable for display in the call control gadget header field

- Select up to 20 call variables for display in 2 column layout
- Single layout definition currently (v10.5)
- Multiple layouts planned for later release

The screenshot displays the Cisco Finesse Administration interface for managing call variable layouts. The page title is 'Manage Call Variables Layout'. It features a navigation bar with options: 'Call Variables Layout' (selected), 'Desktop Layout', 'Phone Books', 'Reasons', 'Team Resources', and 'Workflows'. The main content area is divided into three sections:

- Call Header Layout:** A table with two columns: 'Display Name' and 'Variable'. It contains one row: 'Call Variable 1' with 'callVariable1' selected in the dropdown.
- Call Body Left-Hand Column Layout:** A table with three columns: 'Display Name', 'Variable', and 'Delete?'. It lists 6 rows of variables, each with a red 'X' in the 'Delete?' column. The variables are: BA AccountNumber, BA Campaign, Call Variable 1, Call Variable 2, Call Variable 3, and Call Variable 4.
- Call Body Right-Hand Column Layout:** A table with three columns: 'Display Name', 'Variable', and 'Delete?'. It lists 6 rows of variables, each with a red 'X' in the 'Delete?' column. The variables are: BA Status, BA Response, Call Variable 6, Call Variable 7, Call Variable 8, and Call Variable 9.

At the bottom of each table is an 'Add Row' button. At the bottom of the entire configuration area are 'Save' and 'Revert' buttons.

# Call Variable Data Presentation

The image illustrates the configuration and presentation of call variables in Cisco Finesse. It is divided into two main sections: the administration interface and the agent desktop output.

**Administration Interface (Top):** The 'Manage Call Variables Layout' page shows three configuration areas:

- Call Header Layout:** A table with columns 'Display Name' and 'Variable'. It contains one row: 'Attendee ID' mapped to 'callVariable3'.
- Call Body Left-Hand Column Layout:** A table with columns 'Display Name', 'Variable', and 'Delete?'. It contains three rows: 'Name' (callVariable1), 'Company' (callVariable2), and 'Mobile' (callVariable4). Each row has a red 'X' in the 'Delete?' column.
- Call Body Right-Hand Column Layout:** A table with columns 'Display Name', 'Variable', and 'Delete?'. It contains three rows: 'Attendee ID' (callVariable3), 'Breakout Type' (callVariable5), and 'Session ID' (callVariable6). Each row has a red 'X' in the 'Delete?' column.

**Agent Desktop Output (Bottom):** The interface for 'Agent Ann Dronicus (andronicus) - Extension 7702' shows the following data:

- Call duration: 00:04
- Attendee ID: 106717 (highlighted with an orange box)
- Number: 07260013295
- Buttons: Keypad, Hold, Consult, End
- Call Information Card (Left):
  - Name: Isaac Mason
  - Company: Boyle and Lancet
  - Mobile: 07918398436
- Call Information Card (Right):
  - Attendee ID: 106717
  - Breakout Type: BRKCCT
  - Session ID: 2559
- Buttons: Wrap-Up Reason, Apply

Arrows indicate the flow of data from the configuration tables to the desktop output. An orange arrow points from the 'Attendee ID' in the header layout to the 'Attendee ID' in the desktop output. A blue arrow points from the 'Name', 'Company', and 'Mobile' rows in the left-hand column layout to the corresponding fields in the desktop output. A green arrow points from the 'Attendee ID', 'Breakout Type', and 'Session ID' rows in the right-hand column layout to the corresponding fields in the desktop output.

Resultant Desktop Output

# Workflows

- Define **automated actions** and the **events / conditions** that trigger them
- Each **workflow definition** comprises 3 things –
  1. One of the following triggering events
    - Call arrives
    - Call answered
    - Call ends
    - Making outgoing call
    - Outbound call preview
  2. Conditions that must match for the action to be invoked
    - Match ANY or ALL conditions in list
    - Variable <operator> Value (<operator> is equal, not equal, contains, is in list, etc)
  3. The predefined action that will be performed

# Workflows

### Manage Workflows

#### List of Workflows

Name	Description
Cisco Live Incoming	CC breakout helpline incoming caller details
Log Call Details	Log answered call to back-end call history

### Edit Workflow

Name:

Description:

When to perform Actions:

How to apply Conditions:

✖

#### Ordered List of Actions

Name	Type
Get Caller DB Record	BROWSER_POP

**Trigger Event**

**Conditions**

**Actions List**

# Workflow Actions

## Two Types of Action

1. Browser window pop-up
    - Or, new tab (depending on browser behaviour settings)
    - Re-uses window/tab when same action is next triggered
  2. Make HTTP request
    - POST or PUT method
    - URL either points to Finesse server itself or other external destination
- URLs and request body contain substitution tags to include run-time variables
  - Actions handled by built-in Finesse handler or custom gadget action handler
  - Single action can be used to trigger processing in (multiple) custom gadgets (Gadgets subscribe to workflow action events)

# Workflows

## Browser Pop

### Manage Workflow Actions

**List of Actions**

Name	Type
Add To Call History	HTTP Request
Get Caller DB Record	Browser Pop
Start Recording	HTTP Request

### Edit Action

Name:

Type:

Handled by:

Window Name:

Browser URL:

**Preview**

**Sample Data**

callVariable3	111222
fromAddress	07788996655

Browser URL:

**Annotations:**

- Orange box: Open or re-use window/tab (points to Type and Handled by)
- Green box: Variable substitution and preview (points to callVariable3 in URL and Sample Data)

# Workflows

## Browser Pop

The screenshot shows the 'Manage Workflow Actions' interface. A 'List of Actions' table is visible, with a browser window overlaid on top. The browser window title is 'Tech Breakouts' and the address bar shows the URL: `https://10.52.202.40:8181/custdb?attendeelD=106717&callingNumber=073816449`. The browser content displays 'Cisco Live Milan Contact Centre Tech Breakouts'.

The 'Edit Action' form is shown with the following fields:

- Name: Get Caller DB Record
- Type: Browser Pop
- Handled by: Finesse Desktop
- Window Name: Caller\_Lookup
- Browser URL: http://10.52.202.40:8080/custdb?attendeelD=111222&callingNumber=07788996655

Under the 'Preview' section, 'Sample Data' is shown:

- callVariable3: 111222
- fromAddress: 07788996655

The Browser URL field at the bottom is: `http://10.52.202.40:8080/custdb?attendeelD=111222&callingNumber=07788996655`

The form contains the following fields:

- Name: Isaac Mason
- Mobile: 07918398436
- Email: isaacmason@boylelan
- Company: Boyle and Lancet
- Job Title: Solution Designer
- Attendee ID: 106717

Checkboxes for training modules:

- BRKCCT-2557 Exploiting Features for Maximum Effect in CCX
- BRKCCT-2558 Mobile Devices and Maximising Customer Engagement
- BRKCCT-2559 The Essential Quick Start Guide To Adopting Finesse
- BRKCCT-3005 Solution Troubleshooting for Unified Contact Center Enterprise

Buttons: Update, Cancel

But, how do we get this to pop in a Finesse tab?

1. Set "Handled by" to Other
2. Add a custom gadget to process the workflow action event

# Workflows

## HTTP Request

### Manage Workflow Actions

#### List of Actions

Name	Type
Add To Call History	HTTP Request
Get Caller DB Record	Browser Pop
Start Recording	HTTP Request

### Edit Action

Name:

Type:

Handled by:

Method:

Location:

Content Type:

URL:

Body: 

```
<Dialog>
  <requestedAction>START_RECORDING</requestedAction>
  <targetMediaAddress> extension ✕ </targetMediaAddress>
</Dialog>
```

**HTTP request to Finesse API**

**Variable substitution**

# Workflows

## HTTP Request

Manage Workflow Actions

List of Actions

Name	Type
Add To Call History	HTTP Request
Get Caller DB Record	Browser Pop
Start Recording	HTTP Request

Edit Action

Name: Add To Call History

Type: HTTP Request

Handled by: Finesse Desktop

Method: POST

Location: Other

Content Type: <-www-form-urlencoded

URL: http://10.52.202.40:8080/LogCall

Body: agent= loginId &from= fromAddress &ext= extension

HTTP request to external URL via Finesse server

Request body built with variable substitution

# Workflows

## HTTP Request

### Manage Workflow Actions

#### List of Actions

Name	Type
------	------

```
Internet Protocol, Src: 10.52.200.165 (10.52.200.165), Dst: 10.52.202.40 (10.52.202.40)
Transmission Control Protocol, Src Port: 53536 (53536), Dst Port: http-alt (8080), Seq: 1, Ack: 1, Len: 296
Hypertext Transfer Protocol
  POST /LogCall HTTP/1.1
    Request Method: POST
    Request URI: /LogCall
    Request Version: HTTP/1.1
    Content-Type: application/x-www-form-urlencoded
    X-Forwarded-For: 10.60.183.84
    X-shindig-dos: on
    Content-Length: 42
    Host: 10.52.202.40:8080
    Connection: Keep-Alive
    User-Agent: Apache Shindig
    Accept-Encoding: gzip, deflate
    \r\n
Line-based text data: application/x-www-form-urlencoded
agent=andronicus&from=07674530988&ext=7702
```

URL

Body

Request body built with variable substitution

Cisco *live!*

# Workflows

## HTTP Request

### Manage Workflow Actions

#### List of Actions

Name	Type
------	------

```
Internet Protocol, Src: 10.52.200.165 (10.52.200.165), Dst: 10.52.202.40 (10.52.202.40)
Transmission Control Protocol, Src Port: 53536 (53536), Dst Port: http-alt (8080), Seq: 1, Ack: 1, Len: 296
Hypertext Transfer Protocol
  POST /LogCall HTTP/1.1
    Request Method: POST
    Request URI: /LogCall
    Request Version: HTTP/1.1
    Content-Type: application/x-www-form-urlencoded
    X-Forwarded-For: 10.60.183.84
    X-shindig-dos: on
    Content-Length: 42
    Host: 10.52.202.40:8080
    Connection: Keep-Alive
    User-Agent: Apache Shindig
    Accept-Encoding: gzip, deflate
    \r\n
Line-based text data: application/x-www-form-urlencoded
agent=andronicus&from=07674530988&ext=7702
```

HTTP Body with substituted values

URL `http://10.52.202.40:8080/LogCall`

Body `agent= loginId &from= fromAddress &ext= extension`

Request body built with variable substitution

Cisco *live!*

# Workflows

## HTTP Request

### Manage Workflow Actions

#### List of Actions

```
Internet Protocol, Src: 10.52.200.165 (10.52.200.165), Dst: 10.52.202.40 (10.52.202.40)
Transmission Control Protocol, Src Port: 53536 (53536), Dst Port: http-alt (8080), Seq: 1, Ack: 1, Len: 296
Hypertext Transfer Protocol
  POST /LogCall HTTP/1.1
    Request Method: POST
    Request URI: /LogCall
    Request Version: HTTP/1.1
    Content-Type: application/x-www-form-urlencoded
    X-Forwarded-For: 10.60.183.84
    X-shindig-dos: on
    Content-Length: 42
    Host: 10.52.202.40:8080
    Connection: Keep-Alive
    User-Agent: Apache Shindig
    Accept-Encoding: gzip, deflate
    \r\n
  Line-based text data: application/x-www-form-urlencoded
    agent=andronicus&from=07674530988&ext=7702
```

CCX/Finesse IP Addr: 10.52.200.165  
Web Server IP Addr: 10.52.202.40  
Desktop IP Addr: 10.60.183.84

HTTP Body with substituted values

URL `http://10.52.202.40:8080/LogCall`

Body `agent= loginId &from= fromAddress &ext= extension`

Request body built with variable substitution

Cisco *live!*

# Workflows

## HTTP Request

### Manage Workflow Actions

#### List of Actions

```
Internet Protocol, Src: 10.52.200.165 (10.52.200.165), Dst: 10.52.202.40 (10.52.202.40)
Transmission Control Protocol, Src Port: 53536 (53536), Dst Port: http-alt (8080), Seq: 1, Ack: 1, Len: 296
Hypertext Transfer Protocol
  POST /LogCall HTTP/1.1
    Request Method: POST
    Request URI: /LogCall
    Request Version: HTTP/1.1
    Content-Type: application/x-www-form-urlencoded
    X-Forwarded-For: 10.60.183.84
    X-shindig-dos: on
    Content-Length: 42
    Host: 10.52.202.40:8080
    Connection: Keep-Alive
    User-Agent: Apache Shindig
    Accept-Encoding: gzip, deflate
    \r\n
Line-based text data: application/x-www-form-urlencoded
agent=andronicus&from=07674530988&ext=7702
```

CCX/Finesse IP Addr: 10.52.200.165  
Web Server IP Addr: 10.52.202.40  
Desktop IP Addr: 10.60.183.84

Web server receives desktop request  
via Finesse server

HTTP Body with substituted values

URL `http://10.52.202.40:8080/LogCall`

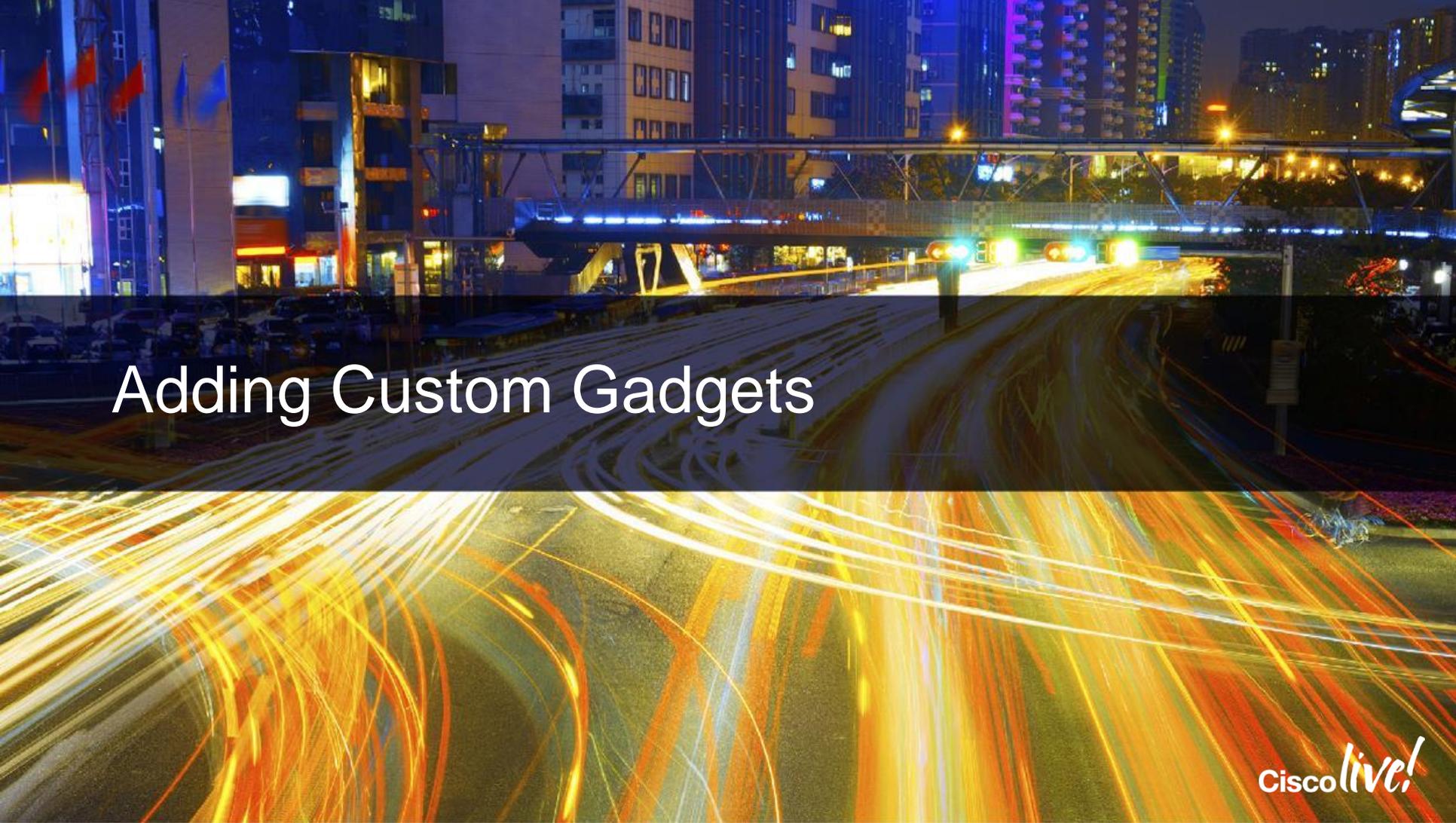
Body `agent= loginId &from= fromAddress &ext= extension`

Request body built with variable substitution

Cisco *live!*

# Workflow Limits

<b>Max Configured Workflows</b>	100
<b>Max Configured Actions</b>	100
<b>Max Conditions Per Workflow</b>	5
<b>Max Actions Per Workflow</b>	5
<b>Max Variable Substitutions Per Action</b>	5
<b>Max Workflows Per Team</b>	20



# Adding Custom Gadgets

# Next Step: From Out-The-Box to Custom Content

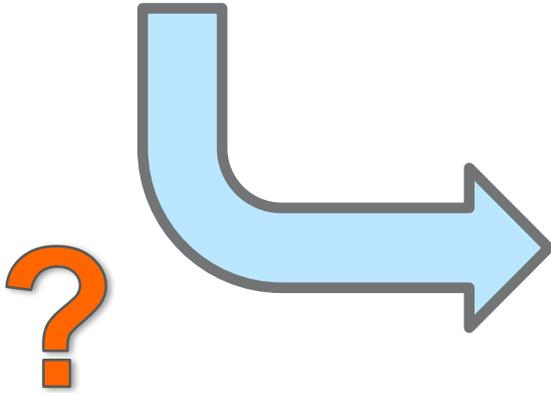
Agent Ann Dronicus (andronicus) - Extension 4410 Sign Out

CISCO Not Ready - Follow-up Actions 00:16

Home Manage Call

Make a New Call

© 2010-2014 Cisco Systems, Inc. All rights reserved. Cisco Finesse v10.5(1)



Agent Ann Dronicus (andronicus) - Extension 4410 Sign Out

CISCO Talking

Home My Statistics Manage Call Charts Layout Map Breakouts Session

00:05 | Attendee ID: 106717 | 07690024280 Keypad Hold Consult End

Name: Isaac Mason Attendee ID: 106717  
Company: Boyle and Lancet Breakout Type: BRKCCCT  
Mobile: 07918398436 Session ID: 2559

Wrap-Up Reason Apply

Security Travel Support

Breakouts

Cisco Live Milan Contact Centre Tech Breakouts

Name: Isaac Mason  
Mobile: 07918398436  
Email: isaacmason@boylelan  
Company: Boyle and Lancet  
Job Title: Solution Designer  
Attendee ID: 106717

Custom Charts - Agent State

Agent State Summary

Agent State	Agent Count
Ready	20
Not Ready	1
Talking	5
Hold	0
After Call Work	0
Work Not Ready	0
Work Ready	0
Reserved	0

Map - Cisco Live Milan

MiCo Milano Congressi

View on Google Maps

MiCo Milano Congressi

©2015 Google - Map data ©2015 Google Terms of Use Report a map error

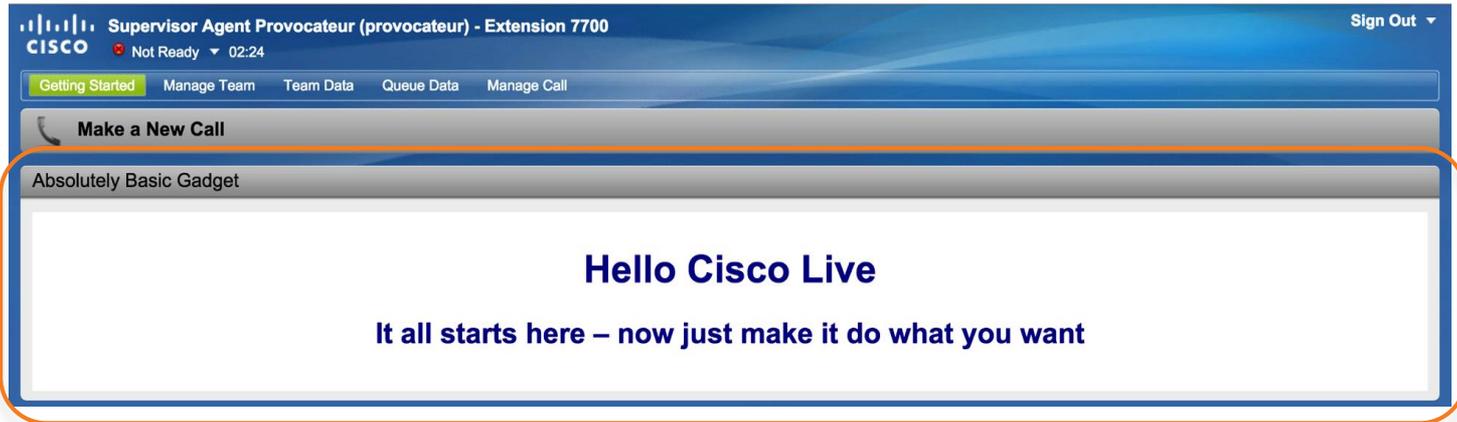
# Making The Leap To Customisation

## Progressively Increasing Level Of Complexity

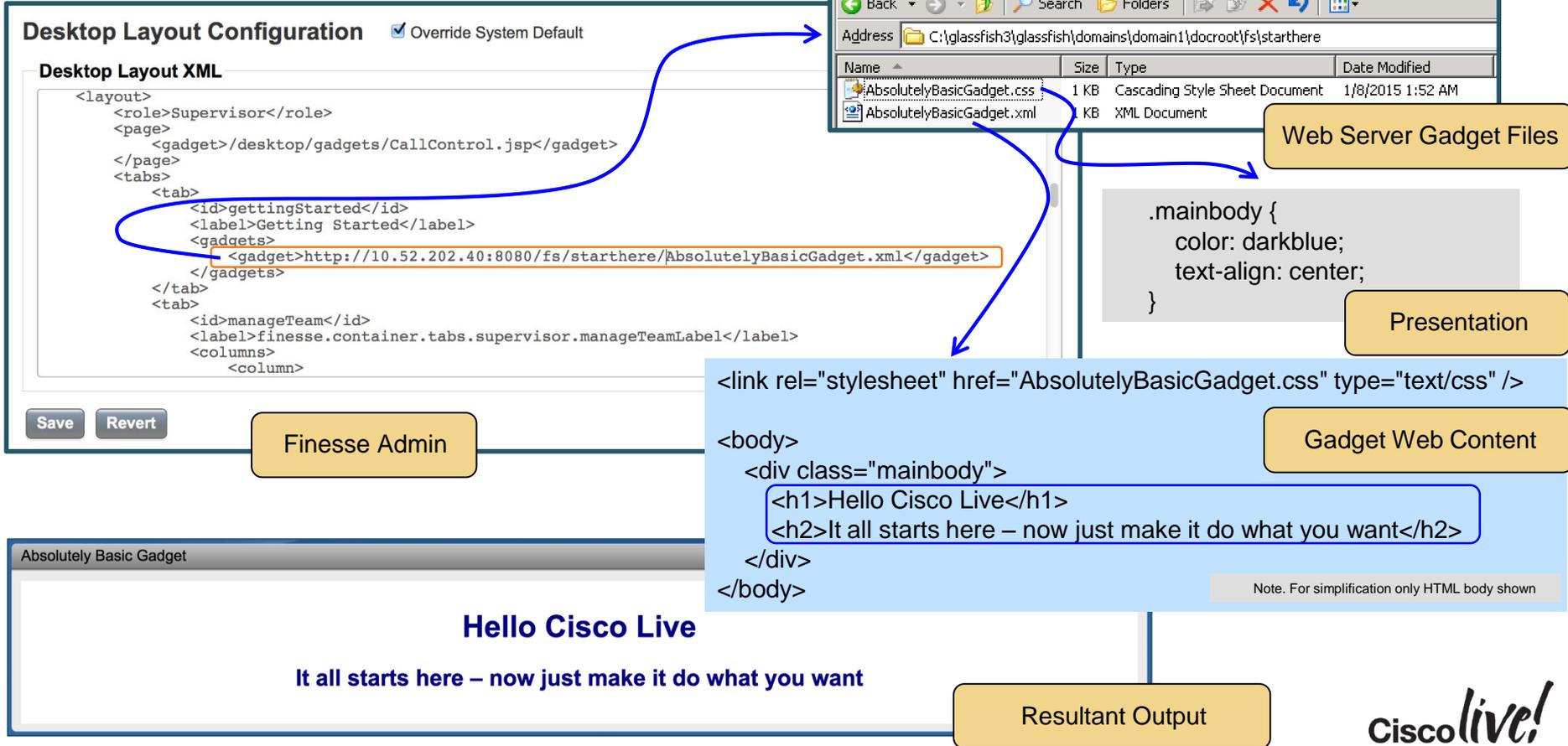
1. Starting out – the “Hello World” gadget
  - Simple passive content using static HTML
2. Insert standalone external web content
  - Content served from external URL, such as:
    - Information tools (weather/time/financial), mapping, reporting data, web applications
3. Actively integrated web content
  - Receiving Finesse events
  - Invoking Finesse API commands
  - Modifying web content dynamically
  - Such as custom buttons, transfer shortcuts, RONA alerts, add custom recording tags

# The “Hello World” Gadget

- Start with something absolutely basic
  - Simply display some passive text
  - Understand the mechanism
  - Build, deploy and add to the desktop



# But, What Do I Actually Build?



# Gadget XML

## Getting started

- Be pragmatic
- Set the gadget title bar value
- Just accept the gadget wrapper XML content
- Understand it later

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Module>
```

```
  <ModulePrefs title="Absolutely Basic Gadget" >
```

```
    <Require feature="settitle" />
```

```
    <Require feature="dynamic-height" />
```

```
    <Require feature="pubsub-2" />
```

```
    <Require feature="setprefs" />
```

```
  </ModulePrefs>
```

```
  <Content type="html">
```

```
<![CDATA[
```

Your Content In Here

```
]]>
```

```
  </Content>
```

```
</Module>
```

# Gadget XML

## Getting started

- Be pragmatic
- Set the gadget title bar value
- Just accept the gadget wrapper XML content
- Understand it later

## Concentrate on the content

- The part within the CDATA section
- That's what will form the rendered gadget content

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Module>
```

```
  <ModulePrefs title="Absolutely Basic Gadget" >
```

```
    <Require feature="settitle" />
```

```
    <Require feature="dynamic-height" />
```

```
    <Require feature="pubsub-2" />
```

```
    <Require feature="setprefs" />
```

```
  </ModulePrefs>
```

```
  <Content type="html">
```

```
    <![CDATA[
```

```
      <link rel="stylesheet" href="AbsolutelyBasicGadget.css" type="text/css" />
```

```
      <body>
```

```
        <div class="mainbody">
```

```
          <h1>Hello Cisco Live</h1>
```

```
          <h2>It all starts here – now just make it do what you want</h2>
```

```
        </div>
```

```
      </body>
```

```
    ]]>
```

```
  </Content>
```

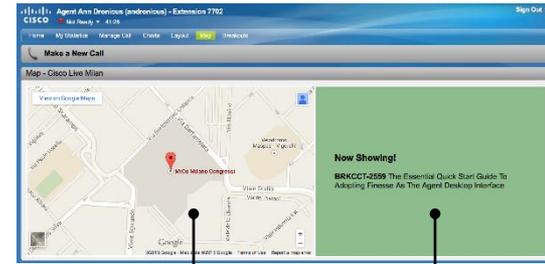
```
</Module>
```



# Embedding Web Content

# Embedded Web-Page / Web-App

- Add an HTML `<iframe>` tag pointing to the required URL
- For example, embedding a Google map in a simple 2 cell table –



```
<body class="claro">
  <table>
    <tr>
      <td>
        <iframe width="600" height="350" frameborder="0" style="border:0"
          src="https://www.google.com/maps/embed/v1/place?zoom=15&q=mico+milan&key=AlzaSyA3...">
        </iframe>
      </td>
      <td class="mapNotes">
        <h3>Now Showing!</h3>
        <b>BRKCCT-2559</b>
        The Essential Quick Start Guide To Adopting Finesse As The Agent Desktop Interface
      </td>
    </tr>
  </table>
</body>
```

# Gadget Sizing Problem

What I want

...

The screenshot shows the Cisco Live agent interface for 'Agent Ann Dronicus (andronicus) - Extension 7702'. The interface includes a top navigation bar with 'Home', 'My Statistics', 'Manage Call', 'Charts', 'Layout', 'Map', and 'Breakouts'. Below this is a 'Make a New Call' button. The main area is titled 'Map - Cisco Live Milan' and contains two gadgets: a map gadget showing a street view of Milan with a red pin at 'MiCo Milano Congressi', and a text gadget with the text 'Now Showing! BRKCT-2559 The Essential Quick Start Guide To Adopting Finesse As The Agent Desktop Interface'. The text gadget is significantly larger than the map gadget.

This screenshot is similar to the one above, but the text gadget is much smaller, making the map gadget more prominent. The text gadget now only displays 'Now Showing!'.

What I get ...

Cisco *live!*

# Getting The Size Correct

- The map iframe is constrained by the size of an outer iframe that contains it
- Need to use JavaScript
- Call **gadgets.window.adjustHeight()** method to resize the container
  - As part of onConnect processing
    - Have to reload for change to become visible
  - When the tab is navigated to
    - Sizing correct but content is rendered in a smaller window resulting in the map incorrectly centered
  - When the tab is navigated to, dynamically insert the iframe into the table and resize
    - Works correctly, best approach

# Re-sizing The Gadget

1. In this example, leave the table cell empty  
(could alternatively use a `<div>` as placeholder for content)
  - assign an ID to be used as a placeholder (`id="loadMapHere"`)
  - add a user-defined attribute as a flag (`data-loaded="false"`)
2. Invoke JavaScript from HTML via `onConnect` handler
3. In JavaScript, register a `Tab-Visible` handler
4. When tab becomes visible and the handler is called, modify the table to include the embedded web-content `iframe`
5. Resize the gadget with the `gadgets.window.adjustHeight()` method

# Successful Embedding – The HTML

```
<script type="text/javascript" src="jquery-1.9.1.min.js"></script>  
<script type="text/javascript" src="finesse-10.5.1.js"></script>  
<script type="text/javascript" src="map.js"></script>
```

Added standard and custom JavaScript

```
<link rel="stylesheet" href="map.css" type="text/css" />
```

```
<body class="claro">
```

```
  <table>
```

```
    <tr>
```

```
      <td id="loadMapHere" data-loaded="false" />
```

Table cell 1 empty, tagged as our placeholder

```
      <td class="mapNotes">
```

```
        <h3>Now Showing!</h3>
```

```
        <b>BRKCCT-2559</b>
```

```
        The Essential Quick Start Guide To Adopting Finesse As The Agent Desktop Interface
```

```
      </td>
```

```
    </tr>
```

```
  </table>
```

```
</body>
```

Set onConnect action to call our custom JavaScript

```
<script type="text/javascript">
```

```
  gadgets.HubSettings.onConnect = function () {finesse.modules.map.init();};  
</script>
```

# Successful Embedding – The JavaScript

```
var finesse = finesse || {};  
finesse.modules = finesse.modules || {};
```

```
finesse.modules.map = (function ($) {  
  var handleTabVisible = function () {  
    if ($("#loadMapHere").attr("data-loaded") == "false")  
    {  
      var url = "https://www.google.com/maps/embed/v1/place?zoom=16&q=mico+milan&key=AlzaSy...";  
      var addhtml = '<iframe width="600" height="350" style="border:0" src="' + url + "'> </iframe>';  
  
      $("#loadMapHere").html(addhtml);  
      $("#loadMapHere").attr("data-loaded", "true");  
  
      gadgets.window.adjustHeight();  
    }  
  };  
};
```

Function called when tab is navigated to

- Tests whether the modification was already done
- Modifies HTML, inserting URL to required web content
- Adjusts the gadget height

```
return {  
  init: function() {  
    containerServices = finesse.containerservices.ContainerServices.init();  
    containerServices.addHandler(finesse.containerservices.ContainerServices.Topics.ACTIVE_TAB, handleTabVisible);  
    finesse.containerservices.ContainerServices.makeActiveTabReq();  
  }  
};  
(jQuery));
```

Function finesse.modules.map.init() called from HTML when gadget is loaded

- Adds the handler to be called on **tab visible events**

# Including Finesse User Credentials

- May be required to include user login credentials on web app URL
- Can include as params on HTTPS URL the iframe points to

```
if ($("#customContent").attr("data-loaded") == "false")
```

```
{
```

```
var prefs = new gadgets.Prefs();  
var cred = finesse.utilities.Utilities.getCredentials(prefs.getString("authorization"));  
usr = cred.id;  
pwd = cred.password;
```

Lookup Finesse user/password

```
var url = "https://10.52.202.40:8181/custdb?user=" + usr + "&password=" + pwd;  
var addhtml = '<iframe width="100%" height="500" style="border:0" src="' + url + '"> </iframe>';
```

Include credentials as URL params

```
$("#customContent").html(addhtml);  
$("#customContent").attr("data-loaded", "true");  
gadgets.window.adjustHeight();
```

Modify gadget content and adjust height

```
}
```

# Modified Gadget Content

The screenshot shows a Cisco web interface with a browser inspector overlay. The interface includes a top navigation bar with 'Home', 'My Statistics', 'Manage Call', 'Charts', 'Layout', 'Map', and 'Breakouts'. Below this is a 'Make a New Call' section. The main content area is titled 'Breakouts' and contains a form with fields for 'Email' (isaacmason@boylelan), 'Company' (Boyle and Lancet), 'Job Title' (Solution Designer), and 'Attendee ID' (106717). A yellow callout box highlights the gadget body ID 'customContent' and the added <iframe> element. A green callout box highlights the network request details.

- Gadget body ID customContent modified
- <iframe> added
- Includes user credentials as URL params

```
<div id="customContent" data-loaded="true">  
  <iframe width="100%" height="500" src="https://10.52.202.40:8181/custdb/?user=andronicus&password=cisco" style="border:0">  
  </iframe>  
</div>
```

Method	File	Domain	Type	Size
GET	https://10.52.202.40:8181/custdb/?user=andronicus&password=cisco			

**Request URL:** https://10.52.202.40:8181/custdb/?user=andronicus&password=cisco  
**Request method:** GET  
**Status code:** 200 OK

HTTPS request for <iframe> content



# Workflow Action Event Handling

# Browser-Pop Into Finesse Gadget

- We've successfully embedded web content into a gadget
- The next stage –
  - Q. How do we refresh it dynamically based on caller data?
  - A. Use a workflow action, Type = **BROWSER\_POP**, Handled By = **OTHER**
- The steps involved –
  1. Enable the gadget to listen for **Workflow Action Events**
  2. Once listening, a gadget receives every workflow action event regardless of type
  3. Analyse the data delivered with the event
  4. Determine whether to process the event or ignore it
  5. Modify the (required section of) gadget HTML with the new URL in the event data  
(Doesn't actually have to use the URL supplied – can do anything whatsoever)

# Listening For Workflow Action Events

```
var finesse = finesse || {};  
finesse.modules = finesse.modules || {};
```

```
finesse.modules.clsession = (function ($) {
```

```
var clientLogger = finesse.cslogger.ClientLogger,
```

```
handleWorkflowAction = function(action) {
```

Event processing in here

```
};
```

```
return {
```

```
init : function () {
```

```
clientLogger.init(gadgets.Hub, "CL-Session-Workflow");
```

```
containerServices = finesse.containerservices.ContainerServices.init();
```

```
clientLogger.log("Adding Workflow-Action handler...")
```

```
containerServices.addHandler(finesse.containerservices.ContainerServices.Topics.WORKFLOW_ACTION_EVENT,  
                             handleWorkflowAction);
```

```
}
```

```
};
```

```
}(jQuery));
```

Function called when workflow action event is received

- Logging enabled so we can see event data
- Especially useful while developing

Function finesse.modules.clsession.init() called from HTML when gadget is loaded

- Adds the handler to be called on **workflow action events**

# Processing The Workflow Action Event

```
handleWorkflowAction = function(action) {
```

```
    var actname = action.getName();  
    var acthandle = action.getHandledBy();  
    var acttype = action.getType();  
    var actparam = action.getParams();
```

Access the action event object data passed to the handler

```
    clientLogger.log("WF Action: " + actname + "/" + acthandle + "/" + acttype);  
    clientLogger.log("Action Params: " + JSON.stringify(action.getParams(), null, 4));
```

Log event data and dump params block

```
    if (acttype == "BROWSER_POP" && acthandle == finesse.containerservices.WorkflowActionEvent.HandledBy.OTHER)  
    {  
        if (actparam.windowName.value == "Session")  
        {
```

Process action if BROWSER\_POP, OTHER, and targeted at window called Session

```
            var wfhtml = '<iframe width="100%" height="500" style="border:0" src="' + actparam.path.expandedValue  
                + "'> </iframe>';  
            $("#customContent").html(wfhtml);  
            gadgets.window.adjustHeight();
```

Modify gadget HTML content

```
        }  
    }  
};
```

# Workflow Action Event

Agent Ann Dronicus (andronicus) - Extension 7702  
Reserved

Home My Statistics Manage Call Charts Layout Map Breakouts **Session**

Attendee ID: 106717 3908  
07964155980

Name: Isaac Mason  
Company: Boyle and Lancet  
Mobile: 07918398436

Attendee ID: 106717  
Breakout Type: BRKCCT  
Session ID: 2559

Answer

## Edit Action

Name Pop Session Details  
Type Browser Pop  
Handled by Other

Window Name Session

Browser URL `ect/search.ww#loadSearch-searchPhrase=%22{callVariable5} - {callVariable6}%22&searchType=session`

## Preview

### Sample Data

callVariable5 BRKCCT

Call arrival triggers workflow action

Workflow action event sent to gadget

Console - https://10.52.200.165:8445/desktop/container/?locale=en\_US&nocache

Inspector Console Debugger Style Editor Performance Network

Net CSS JS Security Logging Clear

```
"2015-01-21T00:34:39.203 +00:00: : ?.?.?.?: Jan 21 2015 00:34:35.496 +0000: CL-Session-Workflow : WF Action Pop Session Details/OTHER/BROWSER_POP" concat:15691
"2015-01-21T00:34:39.203 +00:00: : ?.?.?.?: Jan 21 2015 00:34:35.496 +0000: CL-Session-Workflow : Action Params: { concat:15691
"path": {
  "name": "path",
  "value": "https://www.ciscolivemilan.com/connect/search.ww#loadSearch-searchPhrase=%22{callVariable5}-${callVariable6}%22&searchType=session",
  "expandedValue": "https://www.ciscolivemilan.com/connect/search.ww#loadSearch-searchPhrase=%22BRKCCT-2559%22&searchType=session"
}
"windowName": {
  "name": "windowName",
  "value": "Session",
  "expandedValue": "Session"
}
}"
GET https://www.ciscolivemilan.com/connect/search.ww#loadSearch-searchPhrase=%22BRKCCT-2559%22&searchType=session [HTTP/1.1 200 OK 981ms]
```

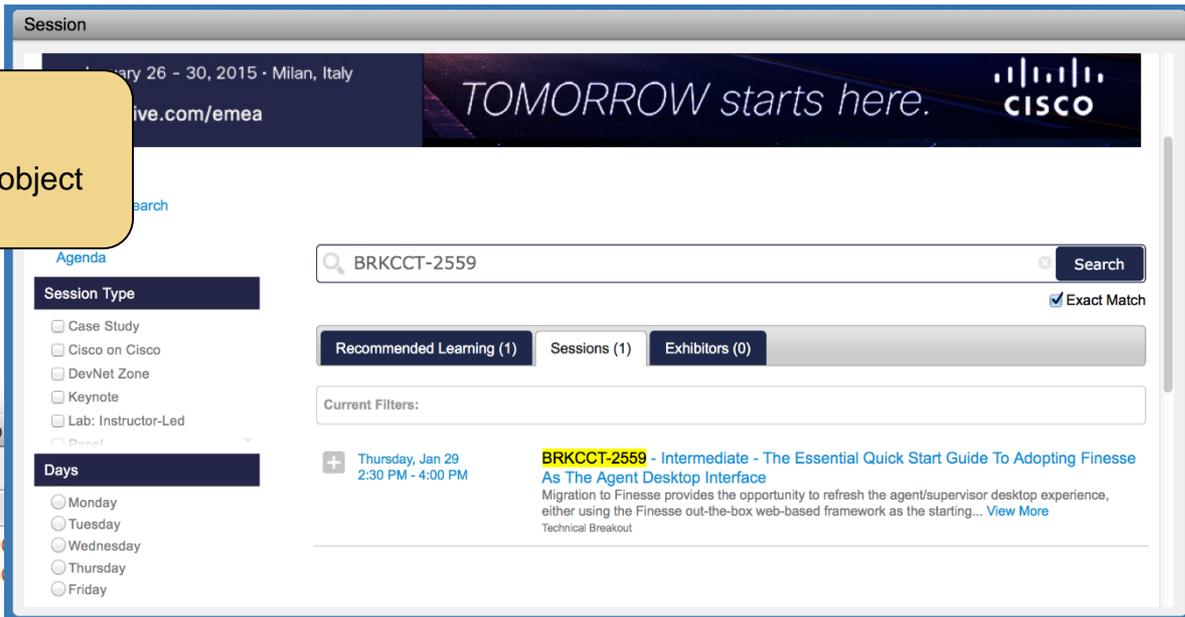
Gadget logging action event data to web console



# Workflow Action Event

- Gadget processes the event
- Inserts modified HTML `<iframe>` tag
- Points to URL provided in the event object
- Finally, URL is loaded by the gadget

```
Console - http
Inspector Console Debugger
Net CSS JS Security Logging
"2015-01-21T00:34:39.203 +00:00: : ?.?.?.?: Jan 21 2015 0
"2015-01-21T00:34:39.203 +00:00: : ?.?.?.?: Jan 21 2015 0
"path": {
  "name": "path",
  "value": "https://www.ciscolivemilan.com/connect/search.ww#loadSearch-searchPhrase=%22%{callVariable5}-${callVariable6}%22&searchType=session",
  "expandedValue": "https://www.ciscolivemilan.com/connect/search.ww#loadSearch-searchPhrase=%22BRKCT-2559%22&searchType=session"
},
"windowName": {
  "name": "windowName",
  "value": "Session",
  "expandedValue": "Session"
}
}"
GET https://www.ciscolivemilan.com/connect/search.ww#loadSearch-searchPhrase=%22BRKCT-2559%22&searchType=session [HTTP/1.1 200 OK 981ms]
```



```
var wfhtml = '<iframe width="100%" height="500" style="border:0" src="' + actparam.path.expandedValue + "'> </iframe>';  
$("#customContent").html(wfhtml);
```

# Using Workflows To Send Data To Gadgets

- Workflow actions can provide a simple way to send commands/data to gadgets
  - Based on workflow trigger conditions
  - Payload comprising up to 5 variables + fixed data values
- Gadgets listen for **Workflow Action Events**
- Filter on Type = **HTTP\_REQUEST**, Handled By = **OTHER**
- Use the request Body as the payload
- Encoding in JSON format means data is accessible as a JavaScript object
- Data fields then conveniently accessed as object properties
  - For example, `payload.targetGadget`, `callInfo.attendeeID`
- Each listening gadget can be coded to process the event/data if required

# Processing The HTTP Request Workflow Action

```
handleWorkflowAction = function(action) {
```

```
  var actname = action.getName();  
  var acthandle = action.getHandledBy();  
  var acttype = action.getType();  
  var actparam = action.getParams();
```

Access the action event object data passed to the handler

```
  clientLogger.log("WF Action: " + actname + "/" + acthandle + "/" + acttype);
```

Process action if it matches HTTP\_REQUEST and OTHER

```
  if (acttype == "HTTP_REQUEST" && acthandle == finesse.containerservices.WorkflowActionEvent.HandledBy.OTHER)  
  {
```

```
    var payload = JSON.parse(actparam.body.expandedValue);
```

Extract data from action event into JS object named payload

```
    clientLogger.log("Body JSON Data: " + JSON.stringify(payload, null, 4));
```

```
    clientLogger.log("Payload Target: " + payload.target);  
    clientLogger.log("Name: " + payload.callerInfo.name);  
    clientLogger.log("Company: " + payload.callerInfo.company);  
    clientLogger.log("Attendee ID: " + payload.callerInfo.attendeelD);
```

Process payload object properties, in this example, just outputting to the log

```
};
```

# HTTP Request Action

**Edit Action**

Name: Send Data To Gadget  
Type: HTTP Request  
Handled by: Other

Method: POST  
Location: Other  
Content Type: application/json  
URL: dummy://

Body: 

```
{ "target": "TestGadget",  
  "callerInfo": {  
    "name": "callVariable1",  
    "company": "callVariable2",  
    "attendeID": "callVariable3" } }
```

Agent Ann Dronicus (andronicus) - Extension 7702  
Home My Statistics Manage Call Charts Layout Map Breakouts Session

Attendee ID: 106717 3908  
07964155980

Name: Isaac Mason  
Company: Boyle and Lancet  
Mobile: 07918398436

Attendee ID: 106717  
Breakout Type: BRKCT  
Session ID: 2559

Answer

Call arrival triggers workflow action

Workflow action event sent to gadget

Action body processed by gadget as JavaScript object

Console - https://10.52.200.165:8445/desktop/container/?locale=en\_US&nocache#breakouts

Inspector Console Debugger Style Editor Performance Network

```
"2015-01-21T23:07:19.281 +00:00 : ?.?.?.?: Jan 21 2015 23:07:19.169 +0000: CL-Session-Workflow : WF Action: Send Data To Gadget/OTHER/HTTP_REQUEST" concat:15691  
"2015-01-21T23:07:19.282 +00:00 : ?.?.?.?: Jan 21 2015 23:07:19.170 +0000: CL-Session-Workflow : Body JSON Data: {  
  "target": "TestGadget",  
  "callerInfo": {  
    "name": "Isaac Mason",  
    "company": "Boyle and Lancet",  
    "attendeID": "106717"  
  }  
}" concat:15691  
"2015-01-21T23:07:19.282 +00:00 : ?.?.?.?: Jan 21 2015 23:07:19.170 +0000: CL-Session-Workflow : Payload Target: TestGadget" concat:15691  
"2015-01-21T23:07:19.282 +00:00 : ?.?.?.?: Jan 21 2015 23:07:19.170 +0000: CL-Session-Workflow : Name: Isaac Mason" concat:15691  
"2015-01-21T23:07:19.282 +00:00 : ?.?.?.?: Jan 21 2015 23:07:19.170 +0000: CL-Session-Workflow : Company: Boyle and Lancet" concat:15691  
"2015-01-21T23:07:19.283 +00:00 : ?.?.?.?: Jan 21 2015 23:07:19.171 +0000: CL-Session-Workflow : Attendee ID: 106717" concat:15691
```

Gadget logging action event data to web console

Payload Target: TestGadget  
Name: Isaac Mason  
Company: Boyle and Lancet  
Attendee ID: 106717

ve!

# Alternative To Monitoring Agent / Call State Changes

- Simpler than monitoring agent state changes and call data
- Allows triggering logic to be configured outside rather than coded into gadgets
- Static commands / data can also be passed in addition to call variables
  - For example, ...
  - Could send a list of transfer shortcut numbers relevant to the specific call type
  - Making dynamic changes to gadget HTML – possibly which fields/options are enabled
- Workflow configuration assigns meaningful names to data items
  - Gadget doesn't have to inspect specific call variables
- Limitations
  - Not all events supported as workflow triggers
  - Max 5 variables can be included in the action event



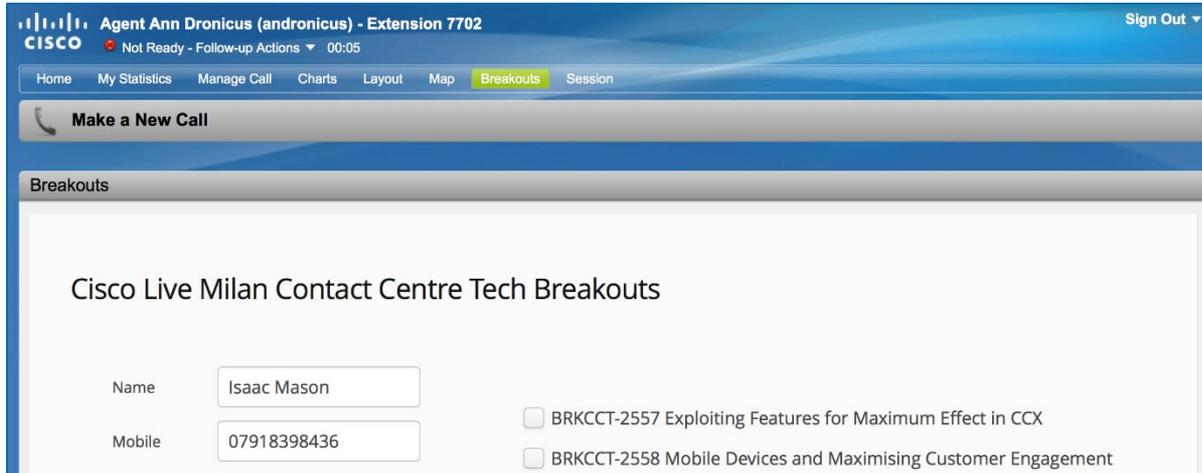
# Call Control And State Change Events

# Call Control And Event Handling

- Gadgetry so far –
  - Using gadgets to include embedded web content
  - Updating web content dynamically
  - Incorporating call data in HTTP requests
  - Handling events resulting from workflow execution
- The next stage –
  - Gadgets performing call control
  - Handling agent and call state events
  - Adding functionality behind agent desktop actions
  - Using the Finesse Client Services API

# Example: Adding Buttons For Transfer Shortcuts

- Use case –
  - Array of custom buttons to function as instant transfer shortcuts
  - Buttons appear only when the agent is in Talking State and transfers are valid



The screenshot displays the Cisco Live agent interface for Agent Ann Dronicus (andronicus) - Extension 7702. The interface includes a navigation bar with options like Home, My Statistics, Manage Call, Charts, Layout, Map, Breakouts, and Session. Below the navigation bar is a 'Make a New Call' button. The main content area is titled 'Breakouts' and contains the text 'Cisco Live Milan Contact Centre Tech Breakouts'. There are two input fields: 'Name' with the value 'Isaac Mason' and 'Mobile' with the value '07918398436'. To the right of these fields are two checkboxes with labels: 'BRKCT-2557 Exploiting Features for Maximum Effect in CCX' and 'BRKCT-2558 Mobile Devices and Maximising Customer Engagement'.

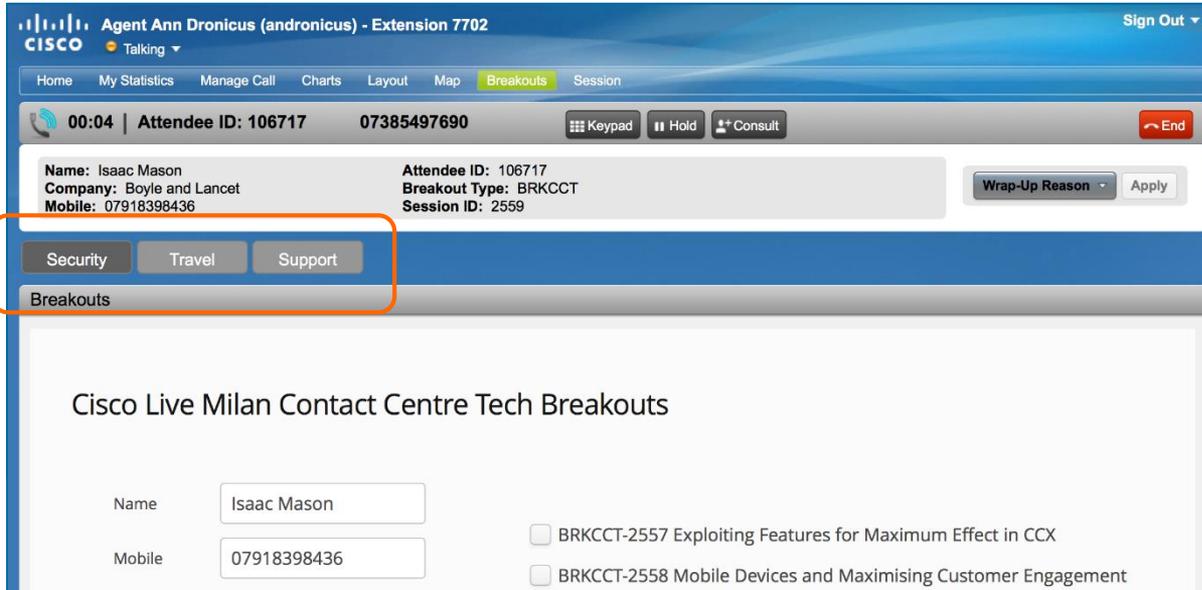
# Example: Adding Buttons For Transfer Shortcuts

- Use case –
  - Array of custom buttons to function as instant transfer shortcuts
  - Buttons appear only when the agent is in Talking State and transfers are valid

The screenshot displays the Cisco Live agent interface for Agent Ann Dronicus (andronicus) - Extension 7702. The interface is in the 'Talking' state. The top navigation bar includes 'Home', 'My Statistics', 'Manage Call', 'Charts', 'Layout', 'Map', 'Breakouts', and 'Session'. The call duration is 00:04, and the Attendee ID is 106717. The phone number is 07385497690. The interface shows a keypad, hold, and consult button. The attendee information is: Name: Isaac Mason, Company: Boyle and Lancet, Mobile: 07918398436. The Breakout Type is BRKCCT and the Session ID is 2559. There is a 'Wrap-Up Reason' dropdown and an 'Apply' button. Below this, there are buttons for 'Security', 'Travel', and 'Support'. The 'Breakouts' section is titled 'Cisco Live Milan Contact Centre Tech Breakouts' and contains two input fields: 'Name' (Isaac Mason) and 'Mobile' (07918398436). There are two checkboxes: 'BRKCCT-2557 Exploiting Features for Maximum Effect in CCX' and 'BRKCCT-2558 Mobile Devices and Maximising Customer Engagement'.

# Example: Adding Buttons For Transfer Shortcuts

- Use case –
  - Array of custom buttons to function as instant transfer shortcuts
  - Buttons appear only when the agent is in Talking State and transfers are valid



The screenshot displays the Cisco Live agent interface for Agent Ann Dronicus (andronicus) - Extension 7702. The interface is in the 'Talking' state. A call is active with a duration of 00:04 and an Attendee ID of 106717. The attendee's name is Isaac Mason, and the company is Boyle and Lancet. The interface shows a 'Breakouts' section with three custom buttons: 'Security', 'Travel', and 'Support', which are highlighted by an orange box and a blue arrow. Below the buttons, there is a 'Wrap-Up Reason' dropdown and an 'Apply' button. The main content area is titled 'Cisco Live Milan Contact Centre Tech Breakouts' and contains a form with fields for Name (Isaac Mason) and Mobile (07918398436). There are also two checkboxes for BRKCCCT-2557 and BRKCCCT-2558.

# Transfer Buttons – The HTML

```
<body class="claro">
  <div id="transfer-block">
    <div id="transfer-buttons-all" class="custom-button-array">
      <div class="custom-button-all">
        <button id="button1" class="custom-button-button"
          onClick="finesse.modules.xferbuttons.transfer(this.id);">1</button>
      </div>
      <div class="custom-button-all">
        <button id="button2" class="custom-button-button"
          onClick="finesse.modules.xferbuttons.transfer(this.id);">2</button>
      </div>
      <div class="custom-button-all">
        <button id="button3" class="custom-button-button"
          onClick="finesse.modules.xferbuttons.transfer(this.id);">3</button>
      </div>
    </div>
    <div align="right" id="statusMsg" class="transfer-status-msg"/>
  </div>
</body>

<script type="text/javascript">
  gadgets.HubSettings.onConnect = function () {finesse.modules.xferbuttons.init();};
</script>
```

### 3 custom buttons with default labels

- Button ID passed to transfer function on click
- Transfer destination could also be configured here but is done in JavaScript in this example

### Classes point to button styling done in CSS

- (Tip) – Use an online button design tool such as [www.cssbuttongenerator.com](http://www.cssbuttongenerator.com)

Transfer status message field populated dynamically by JavaScript

# Transfer Button Processing – The JavaScript

- **init()** – function called when the gadget has loaded
  1. Initialise the Finesse ClientServices interface
    - Supplying hostname / IP and user's access credentials
  2. Add onLoaded and onChange handlers to the user (the agent)
    - To get initial agent information and detect changes in state
  3. Modify the HTML
    - Update the button labels
    - Add the transfer destination to the buttons as user-defined data

# Transfer Button Processing – Initialisation

```
init : function() {  
    ...  
    finesse.clientservices.ClientServices.init(finesse.gadget.Config);  
  
    user = new finesse.restservices.User({  
        id: id,  
        onLoad : userLoaded,  
        onChange : userUpdated  
    });  
  
    $("#button1").text('Security').data('xferdest', '3301');  
    $("#button2").text('Travel').data('xferdest', '3302');  
    $("#button3").text('Support').data('xferdest', '3303');  
}
```

Initialise Finesse ClientServices, passing Config object containing IP addressing and access credentials

Create agent object, getting current status and adding handlers to process the response and notification of updates

Modify the button labels and add the transfer destination numbers as user-defined data

Note. Some JavaScript omitted for simplification

- Button data is set statically in this example
- Alternatively, personal settings could be retrieved from server using agent ID
- Could also be updated dynamically based on call type using a workflow action triggered when a call arrives

# Transfer Button Processing – Agent State

- **userLoaded()** – handler called when user object has loaded
  1. Extract current agent state
  2. Look-up current call dialogs
  3. Add handlers to be called when call dialogs are added and removed
  4. Re-render the gadget display
- **userUpdated()** – handler called when the user state changes
  1. Extract current agent state
  2. Re-render the gadget display

# Transfer Button Processing – Agent State

```
// User load completed so initialise state and do initial rendering of the gadget
```

```
userLoaded = function(evUser) {  
  
    dialogs = evUser.getDialogs( {  
        onCollectionAdd : dialogAdded,  
        onCollectionDelete : dialogRemoved  
    });  
  
    extn = evUser.getExtension();  
    agentState = evUser.getState();  
    updateDisplay();  
},
```

Access the call dialogs object for the agent, adding handlers to be called on new dialogs being added to the dialog collection and existing dialogs being deleted.

Extract the current agent state, save the extension ID for use later in the transfer request, and update the gadget display.

```
// User updated so update state and re-render gadget
```

```
userUpdated = function(evUser) {  
  
    agentState = evUser.getState();  
    updateDisplay();  
},
```

Handler is passed the user object. Simply extract the current agent state and update the gadget display.

# Transfer Button Processing – Call Start / End

- **dialogAdded()** – handler called when a new call dialog starts
  1. Check the current call dialog count
  2. If this is the first one, save it for use later on a transfer request
  3. Add handler to be called when the call dialog state changes
- **dialogRemoved()** – handler called when a call dialog ends
  1. Clear the saved dialog if it's the one that has ended as transfer is no longer an option

# Transfer Button Processing – Call Start / End

```
// Handler for new dialog added

dialogAdded = function(dialog) {

    var dialogCount = dialogs.getDialogCount(true);

    if (dialogCount == 1) {
        xfrdialog = dialog;
    }

    dialog.addHandler('change', callUpdated);
},
```

Look up the number of call dialogs associated with this agent. If this is the first one, then save the dialog for subsequent transfer.

Add a handler to be called on call dialog state changes.

```
// Dialog ended handler

dialogRemoved = function(dialog) {

    if (dialog === xfrdialog) {
        xfrdialog = null;
    }

},
```

Clear our saved dialog if that's the one that has ended. Transfer is no longer possible.

# Transfer Button Processing – Call State

- **callUpdated()** – handler called when the call state has changed
  1. Get the current call state
  2. If the call has failed, update the gadget HTML status field
  3. Update the display to hide the buttons

```
callUpdated = function(dialog) {  
    callState = dialog.getState();  
  
    if (callState === dgstates.FAILED) {  
        $('#statusMsg').html("Call Failed");  
    }  
  
    updateDisplay();  
},
```

Extract current call state

If call failed for some reason during transfer then modify the HTML to show the message and update the button display.

# Transfer Button Processing – Show / Hide

- **updateDisplay()** – function to hide or show the buttons and transfer block
  1. Update the HTML to expose the buttons if the call is in a viable state for transfer
  2. Otherwise hide the buttons and clear the status message

```
updateDisplay = function() {  
  
    if (agentState === agstates.TALKING && callState === dgstates.ACTIVE) {  
  
        $('#transfer-buttons-all').show();  
        $('#transfer-block').show();  
        gadgets.window.adjustHeight();  
  
    } else {  
  
        $('#transfer-block').hide();  
        $('#statusMsg').html("");  
        gadgets.window.adjustHeight(0);  
  
    }  
  
},
```

Call can be transferred if agent is in TALKING state and the call state is ACTIVE, so expose the buttons on the desktop.

# At Last, The Transfer Itself

- **transfer()** – function called when button is pressed
  1. Get the transfer destination from the data attached to the button that was pressed
  2. Perform the transfer by calling the transfer method on the call dialog
  3. Include handlers to be called for success and failure outcomes
  4. Hide the buttons while transfer is in progress and display a progress message

```
transfer : function(button) {  
  if (xfrdialog != null) {  
  
    var target = $("#" + button).data('xferdest');  
  
    xfrdialog.initiateDirectTransfer(extn, target, {  
      success : transferSuccess,  
      error: transferError  
    });  
  
    $('#transfer-buttons-all').hide();  
    $('#statusMsg').html("Transferring to " + target + "...");  
  }  
}
```

Lookup the transfer destination

Invoke single-step transfer (CCE only).  
CCX would require separate consult and  
transfer operations.

Hide buttons and show transfer message

# Handling The Transfer Outcome

- **transferSuccess()** – handler called on transfer executing successfully

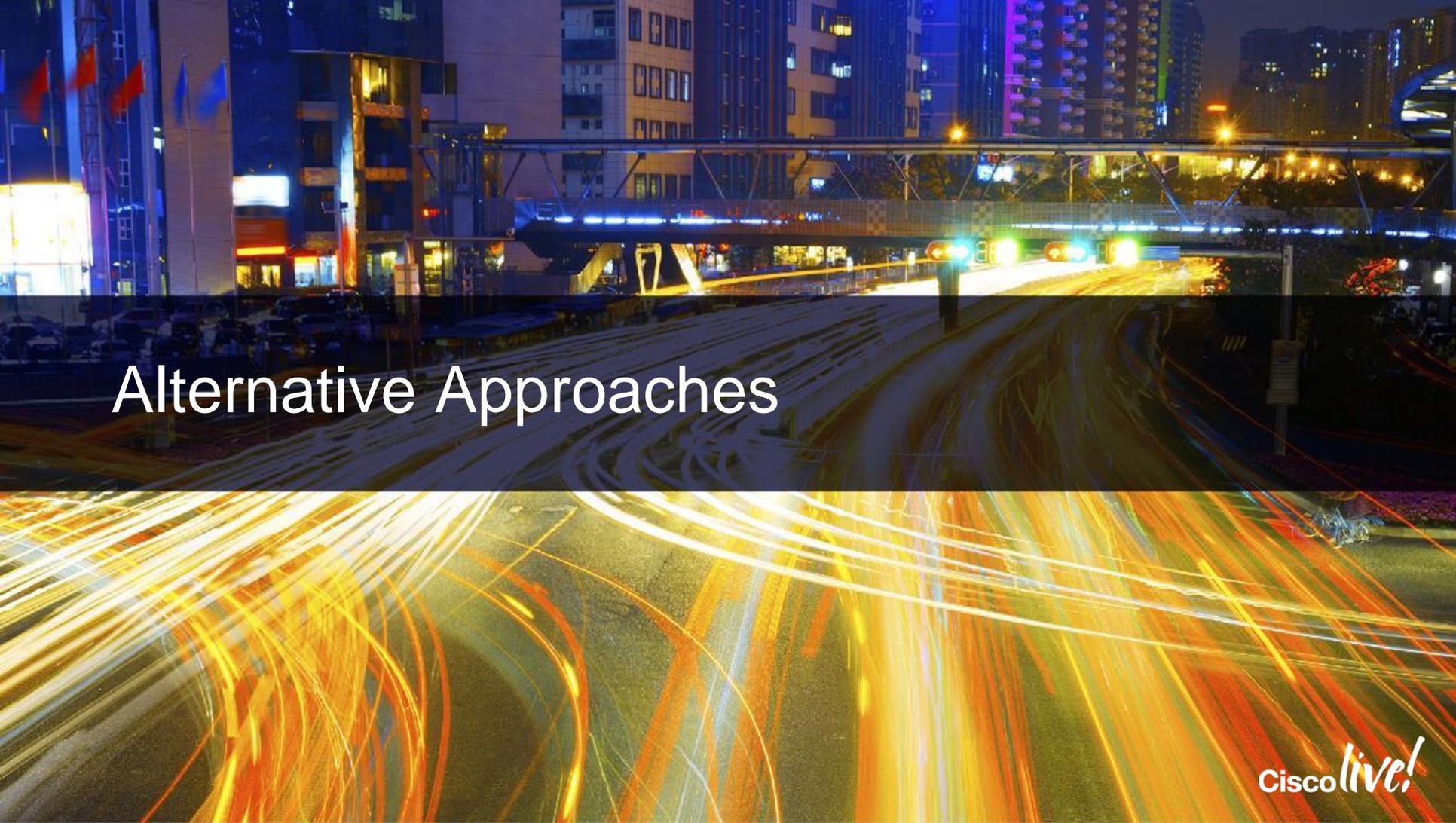
```
transferSuccess = function(rsp) {  
    $('#statusMsg').html("Transferred");  
},
```

Simply display a message to show success

- **transferError()** – handler called when the transfer fails

```
transferError = function(rsp) {  
    $('#statusMsg').html("Transfer Failed");  
    updateDisplay();  
},
```

Display a failure message and update the display to expose the buttons again



# Alternative Approaches

# Direct Integration With Finesse REST API

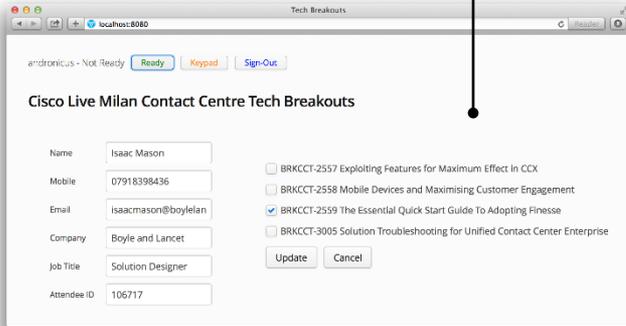
## Possible Use Cases

- Finesse desktop doesn't fit desktop strategy
- Conflicts with business use of desktop real-estate
- Non-browser based agent desktop applications
- Existing desktop application that uses server-side integration
- Custom web application framework using server-side CTI
- Special tools or CTI applications
  - For example, the robot agent simulation described at Cisco Live Milan 2014 (Java application using REST web services client and XMPP listener)
- Any language/environment will work provided HTTP / XMPP messaging possible

# Server-Side Integration With Finesse

## Custom Desktop Web Application

- CTI controls as part of UI only
- Server-side uses Finesse API



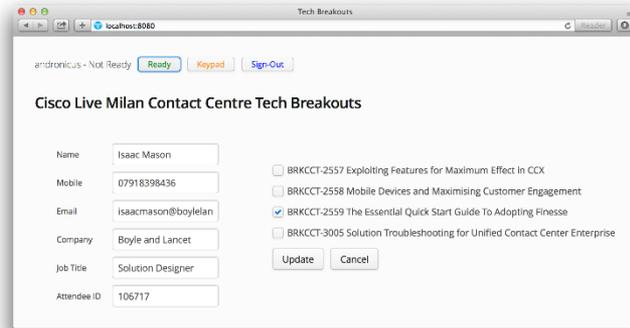
# Client-Side Integration Using JavaScript

## Possible Use Cases

- Existing in-house web-based desktop application
- Third-party web-application if it supports embedding custom web content
- Approach requires embedding additional web content in existing pages
  - Modify page content to include additional CTI desktop controls within the user interface
  - Use JavaScript for Finesse REST API messaging
  - JavaScript XMPP library for event handling
  - Beware the Single Origin Policy
    - One workaround is to use a reverse proxy in front of both Finesse and the application web server
    - Could be located on the web/app server if the application server platform supports it
  - As the starting point, check out the Non-Gadget sample on DevNet

# Browser / Client-Side Integration With Finesse

- JavaScript provides desktop CTI functionality
- Served from web application server
- Browser Single Origin Policy prevents communication back to a different server (Finesse)
- Reverse proxy circumvents restriction by making server-side appear as a single host



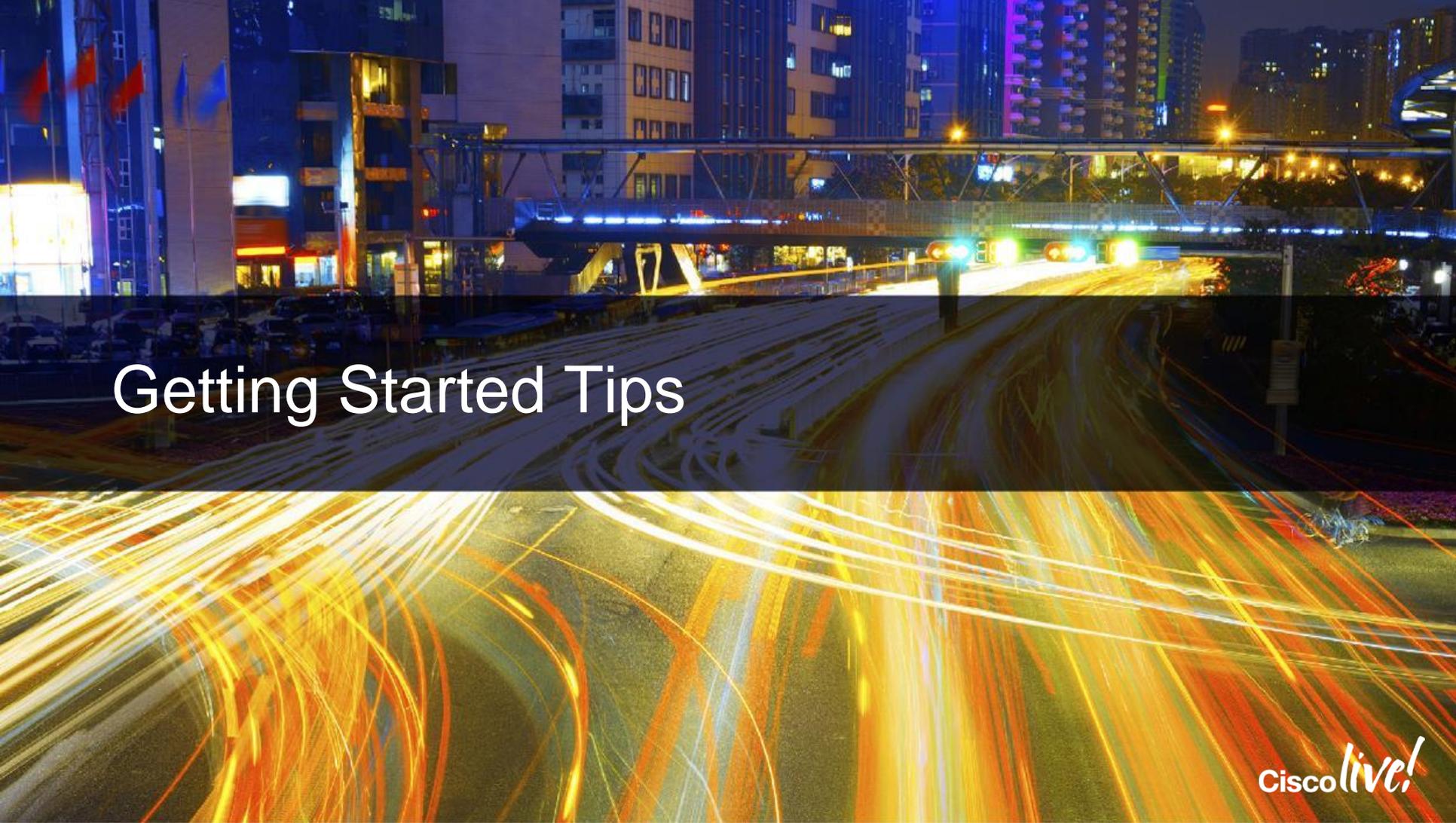
Finesse HTTP REST API  
messaging and XMPP events

Reverse  
Proxy

Web App Content – includes JS for  
client-side interaction with Finesse

Finesse  
Server

Web/App  
Server



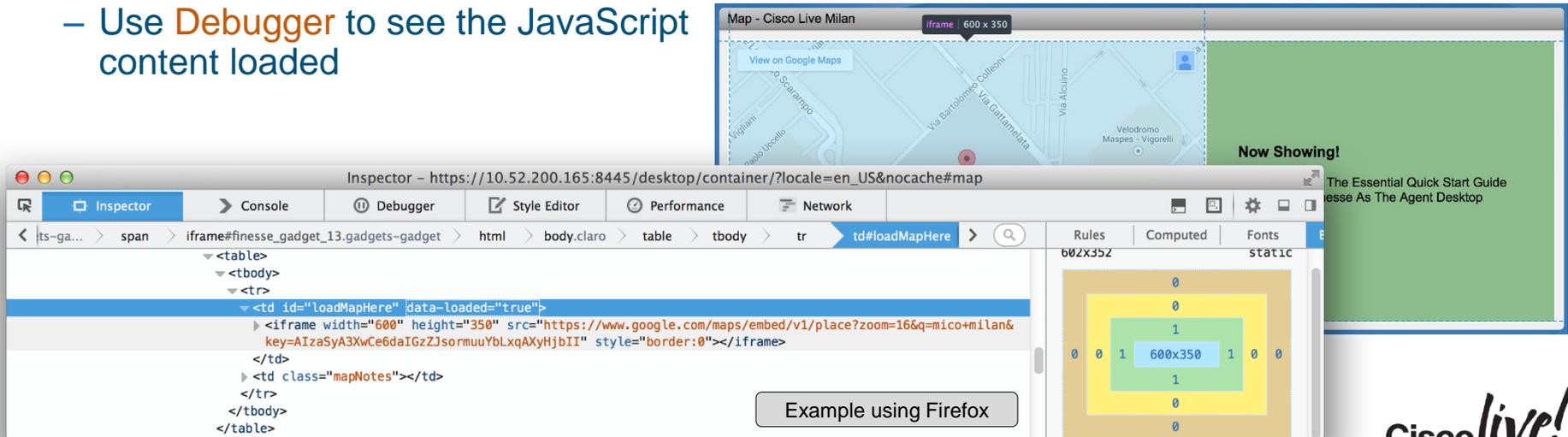
# Getting Started Tips

# A Few Practical Tips

- **Beware The Cache !**
- While developing, add **&nocache** to the Finesse URL
  - For example, [https://fs-dev-svr:8445/desktop/container/?locale=en\\_US&nocache](https://fs-dev-svr:8445/desktop/container/?locale=en_US&nocache)
  - Prevents the Finesse server using cached copies of gadgets and JavaScript
- Clear the browser cache to ensure JavaScript is reloaded
- Ensure external web servers are serving the required versions
  - For example, load the gadget URL directly into a browser to check the XML content

# A Few More Tips

- Make simple use of browser developer tools
- Use the web console to see your diagnostic logging output and run-time errors
- Confirm you've loaded what you think you've loaded
  - Use **Inspect Element** – typically right-click on browser content
  - Can see HTML and CSS styling that's generating the content you're viewing
  - Use **Debugger** to see the JavaScript content loaded



# Final Tips

- Pick a non-complex sample gadget as the starting point for custom work
- Reduce syntactical errors
  - Edit source code using IDE or context sensitive editor
- While developing, use an external web server to host gadgets and associated files
  - Quicker turnaround of changes than loading onto Finesse server
  - Can host on Finesse when completed if preferred
- Have constant supply of generated calls in queue
  - Traffic simulation tools are useful during development and testing
    - Generates calls, which generate events
    - Generates content for calls – reporting data, screen pops based on caller ID
    - See Cisco Live Milan 2014 session BRKCCT-3240 on [ciscolive365.com](https://ciscolive365.com)
    - Materials referenced in that session can be accessed at <https://cisco.box.com/CC-Traffic-Simulation>



Help Is At Hand

Cisco *live!*

# Cisco DevNet For Developer Resources / Samples

Sample Gadgets - Cisco

https://developer.cisco.com/site/collaboration/contact-center/finesse/learn/sample-gadgets/index.gsp

Welcome! Log In Register

DevNet Solution Partner

Menu Overview Documentation Learn Help

HOME > COLLABORATION > CONTACT CENTER > FINESSE > LEARN > SAMPLE GADGETS

## Finesse

### Sample Gadgets

Download a zipped file containing all of these gadgets or download the ones you want below.

### Google Maps Screenpop Sample Gadget 10.0.1

The Google Maps Screenpop gadget provides a screenpop of a Google Maps location from address data in a Callvariable. It requires an API key from Google.

Google Maps Screenpop Gadget

Washington, DC Geocode

Supervisor	Team Name	Total Agents	Logged On	After Call work
11001, Agent	AT11000	9	1	0
11001, Agent	AT11002	9	1	0
11001, Agent	AT11001	9	2	0
		27	4	0

### CUIC Sample Gadget 10.0.1

The CUIC gadget demonstrates how to create a gadget that displays a permalink in an iframe. The gadget is rendered whenever the tab the gadget is on is first visible. (Requires Finesse 9.1(1) or later)

CUIC Gadget

Supervisor	Team Name	Total Agents	Logged On	After Call work
11001, Agent	AT11000	9	1	0
11001, Agent	AT11002	9	1	0
11001, Agent	AT11001	9	2	0
		27	4	0

Download CUICSampleGadget-

### CallRecording Sample Gadget 10.0.1

The CallRecording sample gadget enables a Start Recording button when a call is in progress at the Finesse agent desktop.

Start Recording

Make a New Call

Download the CallRecordingSampleGadget-10.0.1.zip >

### Embedded Salesforce dot com (SFDC) Sample Gadget 10.0.1

EmbeddedSFDC Gadget

Credentials

username: dlender@cisco.com password: \*\*\*\*\* Login

Setme: Developer C

salesforce.com

Home Start Here Merchandise

Create New...

David Lender at Cisco Systems

Calendar

Today 12/11/2013

You have no events scheduled for the next 7 days.

Recent Items

- Finesse Agent1
- Cisco Systems

Download the EmbeddedSFDCSampleGadget-10.0.1.zip >

# Finesse Developer Community Forum

The screenshot shows the Cisco Communities interface for the Finesse Developer Community. At the top, there is a navigation bar with the Cisco logo and 'Communities' text. Below this is a secondary navigation bar with links for 'Products & Services', 'Partners', 'Global', 'Developer', 'Support', 'Learning', and 'Blogs'. The breadcrumb trail indicates the path: 'Cisco Communities > Developer > Collaboration > Contact Center > Finesse'. The main heading is 'Finesse' with a sub-header 'Log in to follow, share, and participate in this community. Not a member? Join Now!'. Below this is a tabbed interface with 'Overview' selected. The main content area contains a welcome message: 'Welcome to the Finesse Community for DevNet! Here you'll find discussions with both developers integrating with Cisco and Cisco internal developers themselves within the Finesse space. Start a discussion or ask a question today! We've migrated the "legacy" Cisco Developer Network (CDN) forum discussions to the documents section of this space.' On the left, there is an 'Actions' sidebar with 'View feeds' and 'Post and Share (Login Required)'. The main content area also features a section titled 'Engage with Finesse Content' with two discussion items: 'UCCE Finesse 10.0 - 3rdpartygadget Error 404' by daniel1435 and 'How can we remove or rename the "Start Monitoring" button in the Team Performance layout?' by FJensen79. To the right, there is a search box for 'Search Finesse' and a 'Top Participants' section.

<https://communities.cisco.com/community/developer/collaboration/contact-center/finesse>

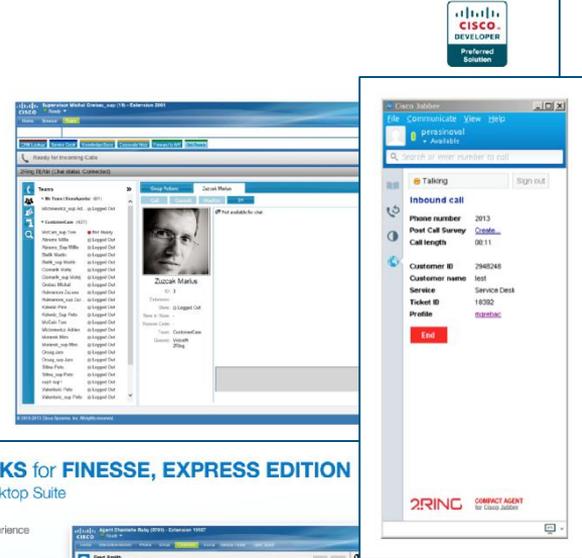
# Development Partner Components

## Bucher+Suter CRM Connectors for Finesse



## 2Ring TEAM

- » Collaborate with other agents based on their presence status
- » Agent Search & Lists (Teams, Queues)
- » Agent Profiles
- » Integrated chat / IM
- » Call / Consult / Monitor
- » Set Ready / LogOut
- » Time in State

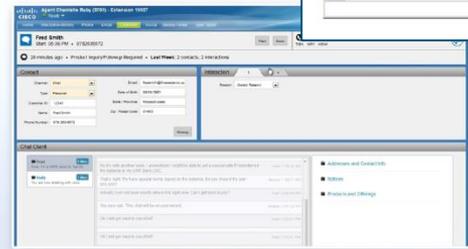


2RING

## UPSTREAM WORKS for FINESSE, EXPRESS EDITION

Omnichannel Finesse Desktop Suite

- Brings sophisticated customer experience "third wave" functionality to UCXC
- Tight integration with UCXC
  - Unified, clean, consistent workflow for all channels
  - Agent management is entirely through native Finesse tools
  - Familiar CAD-like agent to agent chat, and agent/queue statistics included
- Rapid deploy packaged suite extremely simple to install, use and support



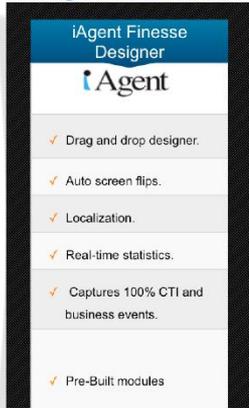
uwf.upstreamworks.com

Preliminary information. Product features subject to change.

Cisco live!

## iAgent

NovelVox



iAgent Finesse Gadget designer provides rapid business application development environment.

# Call to Action

- Visit the World of Solutions for
  - Cisco Campus
  - Walk in Labs
  - Technical Solution Clinics
- Meet the Engineer
- Lunch time Table Topics
- DevNet zone related labs and sessions
- Recommended Reading: for reading material and further resources for this session, please visit [www.pearson-books.com/CLMilan2015](http://www.pearson-books.com/CLMilan2015)

# Complete Your Online Session Evaluation

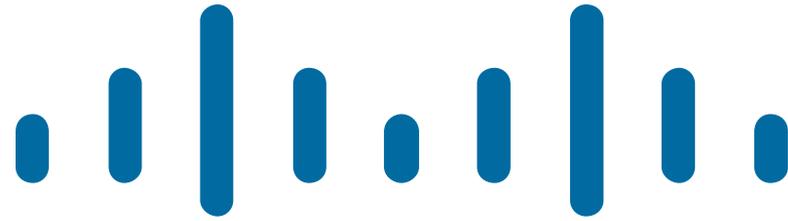
- Please complete your online session evaluations after each session. Complete 4 session evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt.
- All surveys can be completed via the Cisco Live Mobile App or the Communication Stations





Thank you.

Cisco *live!*



**CISCO**