

Glance / Agent Software Integration

*** DRAFT ***

This is a guide for integrating Glance Cobrowse and/or Screenshare with other Agent-side software such as CRM or support ticketing system.

For a diagram of the interaction between the Glance service, Agent software and end-customer, see the [Glance Architecture Overview](#). Basic cobrowse setup is described in the [Cobrowse Setup Guide](#). There is an extensive Javascript API for customizing both the visitor and agent side cobrowse experience that is outside the scope of this document.

This document is arranged as a list of features and requirements to make it easy to create a spec for a specific integration project based on the general information here. Features are noted for a minimal or basic implementation.

Integrating Cobrowse

1. Cobrowse button (minimal)


- A button to join a cobrowse session should be provided, with appropriate branding
- The button should be prominently placed with easy access on UI representing representing people (Lead, Contacts), support cases and other appropriate objects


Examples:

Salesforce Contact - two options for cobrowse button placement

The screenshot displays a Salesforce Contact record for 'Vera Visitor'. At the top, there is a header bar with the contact's name and a 'Glance' button. Below this, the contact's details are shown in a grid format. On the right side of the details grid, there is a 'Glance' button. At the bottom right of the page, there is a 'Glance' button. The 'Glance' button is a blue button with a white 'g' icon and the text 'Cobrowse'.

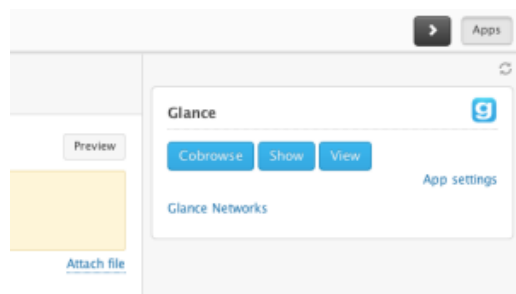
| Title | Account Name | Phone(2) | Email | Contact Owner |
|-------|--------------|--------------|-----------------|---------------|
| | glancevelo | 781-316-2596 | vv@hardebeck.us | Ed Hardeb... |

| RELATED | DETAILS |
|-------------------------------|--|
| Contact Owner Ed Hardebeck |  Cobrowse |
| Name Vera Visitor | Phone 781-316-2596 |

| Glance |
|--|
|  Cobrowse |

| ACTIVITY | CHATTER |
|----------|---------|
|----------|---------|

In Zendesk as a sidebar app next to a ticket:



2. Manual key entry (minimal)


- Input field to type in a cobrowse key that appears to a customer. Normally the key would be read to the agent over a voice call

Examples:

Salesforce


Key:


Zendesk

Glance Cobrowse and Screensharing 

[Glance Networks](#)

Glance Panorama Client:

glance 



3. Visitor ID key from object field (minimal)

When a visitor is identified on the website, some visitor id value can be used as the cobrowse key. The visitor will not need to be presented with a key and the agent will not have to enter it manually.

- Provide the ability to specify for each type of object a standard or custom field in which to find the key
- If field is not empty, do not show input field from 2, use the fieldvalue for cobrowse lookup
- If field is empty, show input field for manual key entry

4. Basic cobrowse join without authentication (minimal)

- Open AgentView url in browser, agent must login again to Glance
- Per-user setting of associated Glance username (optional)

Using the key from 2 or 3, the simplest way to join a cobrowse is to open in a new tab or window the agent join url:

`https://www.glance.net/cobrowse/AgentView.aspx?SessionKey=<key>&wait=1`

If a Glance username is available it should be passed on the url:

`https://www.glance.net/cobrowse/AgentView.aspx?SessionKey=<key>&wait=1&username=<username>`

The agent will need to login with the password for their already provisioned Glance account. The “wait” parameter will enable waiting indication if the session has not yet started on the visitor side.

5. Lookup and waiting indication (basic)

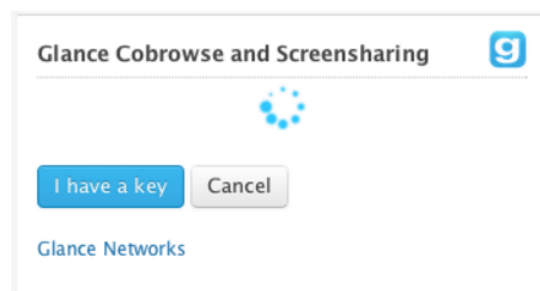
- Call [Lookup](#) web service (from browser JS, or “remoted”)
- Waiting indication while waiting for lookup (session start)
- Ability to cancel waiting
- Open returned AgentView url in browser, agent must login again to glance

A “waiting” UI can be provided in the agent UI, with the ability for the agent to cancel in case the session isn’t started or if the agent typed in the wrong key. A web service is called in a loop, polling for the session to start. If the session is found a URL is returned. This URL would then be opened in a new browser window or tab instead of the URL in 4.

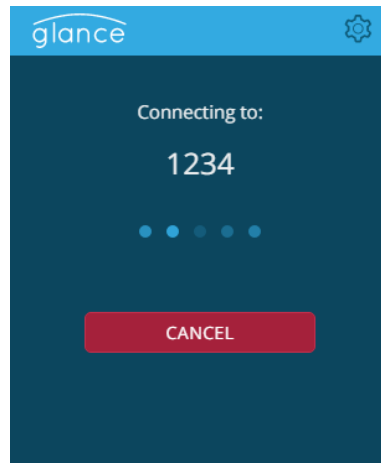
Salesforce

Waiting for guest  [Have key](#) [Cancel](#) 

Zendesk



Glance Panorama Client:



5. Recording of session event / activity (basic)

- Event associated with object form which session is joined
- Start time, key, session type (for future use) added

6. Single-sign-on

- Either Glance SSO ([HMAC token](#) generated from customer id + user id + APIKey)
- Or SAML
- Session lookup using [authenticated web service](#)

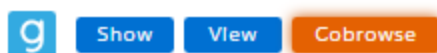
7. Data sync of event details

- Sync between Glance db and agent-side software

Glance has web services that could be used to sync back additional information for session events such as end time, and guest details (location, optional name/email/phone).

8. Presence / Auto-start cobrowse

- Integration of Presence Javascript library / API
- Requires authentication (Glance SSO, SAML could be used)
- Requires visitor id field on object
- Presence of visitor indicated in UI (button lights up, etc)
- Visitor status / page navigation received from JS callbacks, indicated in UI
- Button click to start cobrowse calls JS function
- Callback for Terms and Conditions declined
- Cobrowse join as above



9. Join with Panorama client

- User role based setting for Panorama vs browser join
- Instead of open AgentView url in browser, invoke Panorama client with glancepanorama:// URL (Client is registered as protocol handler)
- Can be authenticated (Glance SSO) or not

Joining a cobrowse session with the Panorama client allows escalation from cobrowse to screenshare, either agent showing their screen, or agent viewing the customer's screen. One way agent video can also be initiated from the Panorama client.

10. Auto-start agent-video

- User role based setting to add parameter to Panorama URL

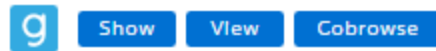
11. Pre-built Reports

- Sessions per agent over time range
- Other...

Integrating Screenshare

Screensharing sessions can be started to Show the agent's screen to the customer, or View the customer's screen, with optional remote control.

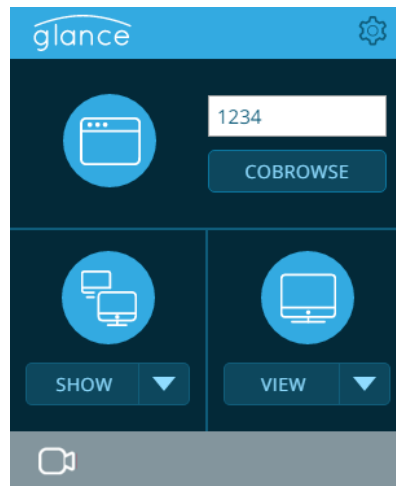
Salesforce



Zendesk



Glance Panorama Client:



Starting screensharing sessions or showing requires the Glance 2.x or 3.x (Panorama) client to be installed or building the Glance SDK into your application.

The client is invoked by opening a URL such as `glance://show?key=1234` (there are many other options documented [here](#)). The client registers itself as a protocol handler. The URL can be opened by software applications in various ways or by a web page in a browser

The simplest integration is to open in an IFRAME this button page:

<https://www.glance.net/integration/HostButtons.asp?username=test.glance.net>



The buttons can be configured and styled in many ways.

A more fully featured screenshare integration in a browser can use a Javascript library. The library can detect if the client is installed, redirect to a download page if it is not, and invoke it with various options. See the [GlanceClientJavascriptAPI](#) document.

13. “View” screenshare

- Invoke installed Glance client to view customer’s screen
- Authentication / event creation
- Per user (or role) based setting for key type (random, fixed, none)

14. “Show” screenshare

- Invoke installed Glance client to show agent’s screen
- Authentication / event creation
- Per user (or role) based setting for key type (random, fixed, none)

Advanced Sync

15. Data-sync of session events for sessions not started via agent software

16. Sync creates Leads for session guests

Some integrations will allow the sync process to create Leads in a CRM. This would be typical in a webinar where guests provide name, phone, email when joining.

History

| | | |
|------------|-----|------------------------------|
| 2016-10-27 | EFH | Initial draft |
| 2016-10-28 | EFH | Expand screensharing section |