Widgets API Reference

Console

2/16/2026

Contents

- Developer

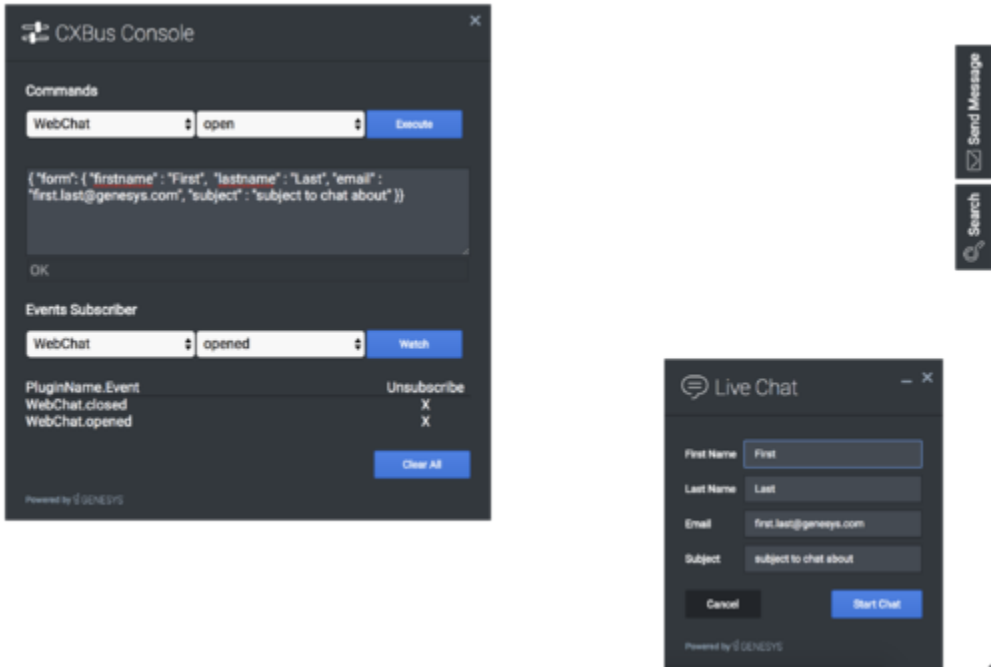
Learn how to debug commands and events on the widget bus.

Related documentation:

-

Overview

Use the Console Widget to debug commands and events on the widget bus. You can use dynamically populated lists to test, debug, or demo all commands. You can also create event watch lists that alert you when an event has fired.



Console provides an easy-to-use interface for debugging the widget bus that complements the standard command-line methods. You can drag and drop the console anywhere on your screen, and when you refresh the page or move to another one, Console reappears right where you left it. It is a great tool for getting to know the widget bus, the API for each widget, and debugging issues.

Usage

Launch WebChat manually by using the following methods:

- Call the **Console.open** command
- Configure the settings to show Console when the browser window is opened.
- Create your own custom button or link to open Console (using the **Console.open** command)

Configuration

Description

Console option to open on initial loading.

Example

```
window._genesys.widgets.console = {open: true};
```

Options

Name	Type	Description	Default	Required
open	boolean	Set to `true` for console to open at start.	false	false

Localization

> **Important**
> For information on how to set up localization, please refer to Localize widgets and services.

Strings

```
{
        "ConsoleTitle": "CXBus Console",
        "Commands": "Commands",
        "Plugin": "Plugin",
        "ConsoleErrorButton": "OK",
        "Execute": "Execute",
        "Event": "Event",
        "SubscribeTo": "Subscribe to",
        "Unsubscribe": "Unsubscribe",
        "ReturnData": "Return Data",
        "EventsSubscriber": "Events Subscriber",
```

```
        "Watch": "Watch",
        "pluginNameEvent": "PluginName.Event",
        "ClearAll": "Clear All",
        "OptionsSample": "JSON Formatted Options {'option': value}"
}
```

API commands

Once you've registered your plugin on the bus, you can call commands on other registered plugins. Here's how to use the global bus object to register a new plugin on the bus.

> ### Important
> The global bus object is a debugging tool. When implementing Widgets on your own site, do not use the global bus object to register your custom plugins. Instead, see Genesys Widgets Extensions for more information about extending Genesys Widgets.

```
var oMyPlugin = window._genesys.widgets.bus.registerPlugin('MyPlugin');

oMyPlugin.command('Console.open');
```

open

Opens the Console UI.

Example

```
oMyPlugin.command('Console.open').done(function(e){

        // Console opened successfully

}).fail(function(e){

        // Console failed to open
});
```

Resolutions

Status	When	Returns
resolved	Console is successfully opened	n/a
rejected	Console is already open	'Already opened'

close

Closes the Console UI.

Example

```
oMyPlugin.command('Console.close').done(function(e){

        // Console closed successfully

}).fail(function(e){

        // Console failed to close
});
```

Resolutions

Status	When	Returns
resolved	Console successfully closed	n/a
rejected	Console is already closed	'Already closed'

configure

Modifies the Console configuration options. See the Console configuration page.

Example

```
oMyPlugin.command('Console.configure', {

        open: false

}).done(function(e){

        // Console configured successfully

}).fail(function(e){

        // Console failed to configure
});
```

Options

Option	Type	Description
open	boolean	If setting is open: true, the console will automatically be open when Widgets is launched and the console is ready.

Resolutions

Status	When	Returns
resolved	Console configuration is provided	n/a
rejected	No configuration is provided	'Invalid Configuration'

API events

Once you've registered your plugin on the bus, you can subscribe to and listen for published events. Here's how to use the global bus object to register a new plugin on the bus.

> **Important**
>
> The global bus object is a debugging tool. When implementing Widgets on your own site, do not use the global bus object to register your custom plugins. Instead, see Genesys Widgets Extensions for more information about extending Genesys Widgets.

```
var oMyPlugin = window._genesys.widgets.bus.registerPlugin('MyPlugin');

oMyPlugin.subscribe('Console.ready', function(e){});
```

Name	Description	Data
ready	Console is initialized and ready to accept commands.	n/a
opened	The Console Widget has appeared on screen.	n/a
closed	The Console Widget has been removed from the screen.	n/a