



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Widget BUS Guide

Widget Bus API Overview

# Widget Bus API Overview

The Widget Bus (CXBUS) is a publish/subscribe/command bus designed for User Interfaces. It allows different UI components and controllers to communicate with each other and bind business logic together into a larger, cohesive product.

CXBUS supports publishing and subscribing arbitrary events with data over the bus and any other plugin on the bus can subscribe to that event. Publications and subscriptions are loosely bound so that you can publish and subscribe to any event without that event explicitly being available. This allows for plugins to lazy load into the bus or provide conditional logic in your plugins that wait for other plugins to be available.

The full CXBUS API reference for each Widget/Plugin is available here: [Widgets Reference](#)

CXBUS events and commands are executed asynchronously using deferred methods and promises. This allows for:

- Better performance
- Standardized Pass/Fail handling for all commands
- Command promises are not resolved until the command is finished, including any nested asynchronous commands that command may invoke. This gives you assurance that the command completed successfully and the timing of your follow-up action will occur at the right time
- Permissions: CXBUS provides metadata in every command call including which plugin called the command and at what time. This allows for plugins to selectively allow/deny invocation of commands.

The Bus is accessible via three methods:

- **Global Access** (Available at runtime after Genesys Widgets have initialized. See **onReady** examples in the [Use Cases](#) below).

`window._genesys.widgets.bus`

This method is very useful for manually triggering commands on the bus using the JavaScript console in your browser. It is also accessible by your own custom programs at runtime for easy integration with Genesys Widgets.

- **Genesys Widgets onReady callback**

```
window._genesys.widgets.onReady = function(CXBUS){  
    // Use the CXBUS object provided here to interface with the bus  
    // CXBUS here is analogous to window._genesys.widgets.bus  
};
```

- **Extensions**

You can define your own plugins/widgets that interface with Genesys Widgets. For more information, please see [Extensions](#).

## Use Cases

### Important

For the following examples let us assume we are working within the **onReady** callback function.

### Example 1: Subscribing to an Event

**Format:** `CXBus.subscribe("event name and path", function(e){})`

**Example:**

```
CXBus.subscribe("WebChat.ready", function(e){  
    // interact with the WebChat widget now that it is initialized (ready)  
});
```

### Example 2: Invoking a Command

**Format:** `widgetBus.command("command name and path", options).done(function(e){}).fail(function(e){})`

**Example:**

```
CXBus.command("WebChat.open").done(function(e){  
    // success scenario  
    // the value of e depends on the command being called  
    // Review the API reference for the particular command you are calling  
}).fail(function(e){  
    // failure scenario: error, exception, improper arguments  
    // the value of e is typically an error string or an AJAX response object  
    // Review the API reference for the particular command you are calling  
})
```