

NBCB V1 Protocol Doc

Nagae Browser Controlling Protocol

V1

ZRY

2023-12-07

V0.1

1 Introduction 介绍

1.1 [O] zh-CN 中文介绍

NBCP是为ProjectNagae项目（商业）设计的协议，设计目的是为了向基于Web的ROV控制系统GUI开发提供后端控制专用浏览器的行为的能力。

由于NBCP相关部分具有较好的泛用性，且与商业项目ProjectNagae的耦合程度很低，因此决定将这一部分作为开源项目发布。与该协议相关的库一同发布的还有一个基于WebView2的简单的开源的专用浏览器NagaeSimpleWebBrowser。

本文档的原始语言为中文，章节标题标记[O] zh-CN的部分为作者直接用中文编写的。

作者可能会机翻或者人工翻译部分章节为英文，人工翻译的部分标记[H] en-US，机翻的部分标记[M] en-US，若翻译内容和原文有冲突或歧义，以[O]标记的为准。

1.2 [M] en-US 英文介绍

NBCP is a protocol designed for ProjectNagae, a business project for ROV controlling system. This protocol is designed for providing a capacity to controlling behaviors of web browser from backend.

NBCP is highly decoupled from ProjectNagae, so I made it open source. With this open source protocol library implementation, there' s also a open source web browser based on WebView2, named NagaeSimpleWebBrowser.

The origin language of this document is zh-CN, with [O] zh-CN marked in the section title.

The author may translate some of sections into English, by machine translating or by manual translating. The title marked with [H] en-US is translated by human, and the title marked with [M] en-US is translated by machine. If there is any inconsistency between the translation and the original text, the original text marked with [O] shall prevail.

2 协议说明

2.1 [O] zh-CN NBCP协议说明

该协议基于WebSocket，后端提供WebSocket接口，浏览器连接至后端的NBCP URL，一般习惯性使用/nbcp，但也可根据实际需要使用其他的路径。

该URL提供的接口为HTTP GET，通过Upgrade升级为WebSocket。

在该接口上通过双向传递json实现通信。

2.1.1 握手过程

1. 浏览器连接WebSocket端点（一般为/nbcp）
2. 如果是新连接，则创建新的会话；如果是因为网络暂时中断导致连接断开，希望尝试继续之前的会话，则重新访问之前的会话。
3. 浏览器与后端建立会话后，后端可向浏览器发起RPC调用（rpc），浏览器可向后端发送消息（client_post）。
4. 后端向浏览器发送会话终止报文（end_session），会话结束。

2.1.2 协议版本

握手时会检验协议版本。版本号分major和minor两部分，只有协议major版本相同的浏览器和后端可以建立会话，对于minor版本，浏览器的协议minor版本需大于等于后端的minor版本。

2.1.3 后端控制浏览器

在会话建立后，后端通过RPC调用，调用浏览器提供的函数实现对浏览器的操作，例如创建浏览器窗口等。

2.1.4 浏览器事件通知

浏览器一端的事件通过client_post向后端发起通知。

2.2 [M] en-US NBCP Protocol Brief

The protocol is based on WebSocket, the backend provides a WebSocket interface, the browser connects to the backend NBCP URL, generally customary to use /nbcp, but can be used according to the actual need to use other path.

The interface provided by the URL is HTTP GET, which is upgraded to WebSocket by Upgrade.

Communication is achieved by passing json in both directions on this interface.

2.2.1 Handshake Process

1. Browser connects to WebSocket endpoint (typically /nbcp)
2. A new session is created if it is a new connection, or the previous session is revisited if the connection was broken due to a temporary network outage and you wish to try to continue the previous session.

3. After the browser establishes a session with the backend, the backend can make an RPC call to the browser (`rpc`) and the browser can send a message to the backend (`client_post`).
4. The backend sends a session termination message (`end_session`) to the browser and the session ends.

2.2.2 Protocol Version

The protocol version is checked during the handshake. The version number is divided into two parts: major and minor. Only browsers and backends with the same major version of the protocol can establish a session, and for minor version, the minor version of the browser's protocol should be greater than or equal to the minor version of the backend.

2.2.3 Backend Controls The Browser

After the session is established, the backend calls the functions provided by the browser through RPC calls to realize operations on the browser, such as creating browser windows.

2.2.4 Browser event notification

Events on the browser side initiate notifications to the backend via `client_post`.

3 报文定义

TODO

4 RPC参考

4.1 [O] zh-CN

协议版本:

- major=1
- minor=1

4.1.1 close_window

关闭指定窗口

参数:

参数名称	数据类型	说明
------	------	----

name	string	窗口名称
------	--------	------

4.1.2 minimize

最小化窗口

参数:

参数名称	数据类型	说明
name	string	窗口名称

4.1.3 new_window

创建新窗口

参数:

参数名称	数据类型	说明
name	string	窗口名称
title	string	窗口标题
url	string	浏览器初始URL
width	int	窗口宽度
height	int	窗口高度
hide_frame	bool	隐藏窗口边框（标题栏）
disable_maximize_btn	bool	禁用最大化按钮
disable_minimize_btn	bool	禁用最小化按钮
disable_close_btn	bool	禁用关闭按钮
maximize_on_show	bool	窗口创建后以最大化显示
no_resizable	bool	窗口大小不可改变

4.1.4 open_dev_tool

打开开发者工具

参数:

参数名称	数据类型	说明
name	string	窗口名称

4.1.5 set_titlebar_display

设置标题栏显示

参数:

参数名称	数据类型	说明
name	string	窗口名称
value	bool	是否显示标题栏

4.1.6 toggle_maximize

最大化/还原窗口

参数:

参数名称	数据类型	说明
name	string	窗口名称

5 事件消息参考

5.1 [O] zh-CN

协议版本:

- major=1
- minor=1

5.1.1 all_window_closed

所有窗口均已经关闭, 一般用来结束会话

参数:

参数名称	数据类型	说明
------	------	----

5.1.2 webview2_core_loaded

窗口内的webview2控件core部分加载完成

参数:

参数名称	数据类型	说明
name	string	窗口名称

5.1.3 window_closed

窗口关闭

参数:

参数名称	数据类型	说明
------	------	----

name	string	窗口名称
------	--------	------

6 NagaeSimpleWebBrowser说明

6.1 [O] zh-CN

该浏览器启动时可通过命令行参数或配置文件指定工作方式。

6.1.1 命令行参数说明

基本格式:

`NagaeSimpleWebBrowser.exe [verb] [options]`

其中verb可以有以下几种:

1. `cfg` 以json配置文件指示启动。不指定verb时这是默认verb。
2. `go-url` 直接打开指定URL的模式
3. `with-nbcp` 以NBCP模式启动

6.1.1.1 `cfg`

在`cfg`模式下有以下选项:

1. `-p` 使用预设文件启动
2. `-f` 指定json文件启动

在程序目录下若存在`cfg-presets`目录, 里面放置扩展名为`.json`的预设文件, 则可以指定`-p`选项并指定json的名称 (不含扩展名)。

例如:

`NagaeSimpleWebBrowser.exe cfg -p foobar`

则会加载`cfg-presets/foobar.json`作为配置文件。

使用`-f`选项可以指定json文件的路径, 加载指定的json文件作为配置文件。例如:

`NagaeSimpleWebBrowser.exe cfg -f "X:\foobar.json"`

以上情况均可以省略verb `cfg`, 例如:

`NagaeSimpleWebBrowser.exe -f "X:\foobar.json"`

配置文件json的示例:

```
{
  "go-url": {
    "url": "http://localhost:4335/",
    "title": "3D Print Cam",
    "maximize-on-show": true
  }
}
```

```
}  
}
```

其中go-url也可以为with-nbcp，具体字段参考go-url和with-nbcp两个verb的参数说明。

6.1.1.2 go-url

参数列表：

参数名称	数据类型	说明
url	string	打开的网页URL
width	int	窗口宽度
height	int	窗口高度
title	string	窗口标题
disable-maximize-btn	bool	禁用最大化按钮
disable-minimize-btn	bool	禁用最小化按钮
maximize-on-show	bool	窗口创建后最大化
no-resizable	bool	禁止调整窗口大小

例如：

```
NagaeSimpleWebBrowser.exe go-url --url "http://www.swzry.com/" --  
maximize-on-show true
```

使用json配置的示例在前面的cfg一节已有示例。

6.1.1.3 with-nbcp

只需要提供一个参数：nbcp的URL，例如：

```
NagaeSimpleWebBrowser.exe with-nbcp --url ws://localhost:9299/nbcp
```

如果采用json配置文件，则是：

```
{  
  "with-nbcp": {  
    "url": "ws://localhost:9299/nbcp"  
  }  
}
```

6.1.2 暴露到前端的接口

NagaeSimpleWebBrowser的WebView2暴露了js对象，可以通过js访问。主要是为后面介绍的Web Remote功能提供必要的信息。

所有接口均位于chrome.webview.host0Objects.nbcp。

major=1 minor=1的版本提供了如下接口：

6.1.2.1 getSessionInfo()

返回值类型为string，获得sessionInfo。

6.1.2.2 getWindowName()

返回值类型为string，获得当前窗口的名称。

6.1.2.3 getNBCPServerUrl()

返回值类型为string，获得当前NBCP端点的URL。

7 Web Remote

7.1 [O] zh-CN

这是提供给前端的接口，可以通过HTTP POST传递JSON来操作，实现在前端上发RPC指令到后端，由后端通过NBCP下发至浏览器。

该接口通常使用/nbcp_web_remote的URL，也可以根据实际情况修改。

7.1.1 nbcp-web-comm

该npm包v0.1.0及以上版本提供了Web Remote接口。

以下提供基于quasar前端框架的示例：

在quasar内添加boot项

nbcp.js:

```
import { boot } from 'quasar/wrappers'
import { NBCPWebComm, replaceWebSocketWithHttp } from "nbcp-web-comm"

export default boot(async ({ app }) => {
  const sessInfo = await
chrome.webview.hostObjects.nbcp.getSessionInfo();
  const winName = await
chrome.webview.hostObjects.nbcp.getWindowName();
  const nbcpBase = await
chrome.webview.hostObjects.nbcp.getNBCPServerUrl();
  const nbcpWebRemote = replaceWebSocketWithHttp(nbcpBase) +
"_web_remote"

  const nbcpWebComm = new NBCPWebComm(sessInfo, winName,
nbcpWebRemote);
  app.config.globalProperties.$nbcp = nbcpWebComm;
})
```

之后便可以在视图内调用了。例如：

App.vue:

```
export default defineComponent({
  name: 'App',
  methods: {
    minimize(){
      this.$nbcpc.rpc("minimize", {"name": this.$nbcpc.getWindowName()})
    },
    closeWindow(){
      this.$nbcpc.rpc("close_window", {"name": this.
$nbcp.getWindowName()})
    },
    closeApp(){
      this.$nbcpc.endSession()
    },
    openDevTool(){
      this.$nbcpc.rpc("open_dev_tool", {"name": this.
$nbcp.getWindowName()})
    },
    setTitleBarDisplay(val){
      this.$nbcpc.rpc("set_titlebar_display", {"name": this.
$nbcp.getWindowName(), "value": val})
    }
  }
})
```