記事

Seisuke Nakahashi · 2021年4月20日 4m read

# VSCode を使ってみよう (2021年4月20日版)

みなさん、こんにちは!

Visual Studio Code 用のエクステンション、 <u>InterSystems Server Manager</u> のバージョン2.0がリリースされました。

このバージョンでは、弊社製品に同梱されているコードエディタ Studio と同じように、「 サーバ上にあるクラス定義や ルーチンを、サーバに接続して、直接編集&コンパイル」することが簡単にできます。この記事では、「Visual Studio Code を、Studio と同じように、気軽に使う」ための設定をご紹介いたします。

# 1. ObjectScript**用エクステンションのインストール**

まずは Visual Studio Code に、InterSystems IRIS (Caché) に接続するためのエクステンションを入れましょう。<u>この記事</u>を参考に、3つのエクステンションを導入します。

- InterSystems ObjectScript
- InterSystems Language Server
- InterSystems Server Manager

# 2. 接続するサーバの登録

左のサイドバーに InterSystems Tools

という道具箱のアイコンがインストールされます。ここで接続先のサーバ情報を定義します。All Servers にカーソルをあわせ + をクリックし、新規接続サーバを登録してください。

#### VSCode を使ってみよう (2021年4月20日版) Published on InterSystems Developer Community (https://community.intersystems.com)



サーバ情報として順に、「接続名、IPアドレス、WEBサーバポート番号、接続ユーザ名、スキーマ (http or https)」を入力していきます。

MS Name of new server definition

en Press 'Enter' to confirm your input or 'Escape' to cancel

erverc

これでサーバ情報がツリーに登録されました!クリックしてサーバに接続すると、接続パスワードを聞かれます。 パスワードを入力し、右上の鍵ボタンをクリックすると、次回以降のパスワード入力をスキップできます。

### VSCode を使ってみよう (2021年4月20日版)

Published on InterSystems Developer Community (https://community.intersystems.com)

INTERSYSTEMS	Password for InterSystems server 'iris2021'			
> 😗 Recent	Password for user '_system' on 'iris2021'			
✓  ☐ All Ser	For To store your password securely, submit it using the $2^{\circ}$ button (Press fat 'Enter' to confirm or 'Escape' to cancel)			
> 且 defau	ult~ensemble			
> 且 defau	ult~iris			
∩ 🚊 iris202	21 ★ ※ 🖸			

### 接続が完了すると、ネームスペース一覧が見えます。



# 3. クラス定義やルーチンを編集する

道具箱 (InterSystems Tools) で接続先を定義しましたが、実際の編集は、Visual Studio Code の Explorer ビューから行います。まず道具箱アイコンから、編集したいネームスペースにカーソルをあわせて、鉛筆マークをクリックしてください。



WORKSPACE上に、以下の画像のように、 **接続サーバ名** : **ネームスペース** 

という名前の論理フォルダが定義されます。ここから、サーバ上のクラス定義やルーチンを直接エディットすることが出来ます。この例のネームスペース NAKA には、まだルーチンもクラスも存在しないので、からっぽです。



(1) ルーチンを新規で作成するには、フォルダ右クリック > New File > 拡張子 mac を入力します。



自由にエディットしてください。Ctrl+S で保存すると自動でコンパイルされます。

≡ test.ma	nc X		ဦဝ	<u>,                                    </u>
1	ROUTINE test			
2	write "これ	は test.mac	です"	
3	quit			

(2) クラスを新規で作成するには、フォルダ右クリック > New File > 拡張子 cls を入力します。



InterSystems Language Server エクステンションによって、キーワードも補完されます。ルーチンと同じく、Ctrl+Sでコンパイルされます。

• User.test.cls			—			×
User.test.cls 3			မို	<u>,                                    </u>		••••
1 2 Class User 3 { 4 Prop	•.test					
Property				Pr	roper	ty
<b>Prop</b> erty		ι	Jniqu	le Pr	roper	ty
<b>Prop</b> erty		Always-Cor	npute	ed Pr	roper	ty
<b>Prop</b> erty		Date	e/Tin	ne Pr	roper	ty

# 4. 接続を定義したワークスペースを保存しておく

次回以降のために、 File > Save Workspace As... でワークスペースを保存しておきましょう ! **ワークスペース名**.code-workspace というファイルで保存されます。

### VSCode を使ってみよう (2021年4月20日版)

Published on InterSystems Developer Community (https://community.intersystems.com)



# 5. CSPファイルを編集する

CSPファイルを編集するには、上記で保存した code-workspaceファイルを、直接編集します。

以下の赤字部分が、CSP編集用の接続定義です。uriはisfs://接続サーバ:ネームスペース/?cspとなります。

### VSCode **を使ってみよう** (2021**年**4月20日版)

Published on InterSystems Developer Community (https://community.intersystems.com)

```
{
    "folders": [
        {
            "name": "iris2021:NAKA",
            "uri": "isfs://iris2021:naka/"
        },
        {
            "name": "iris2021:NAKA:csp",
            "uri": "isfs://iris2021:naka/?csp"
        }
    ]
}
```

### これで、サーバ上のCSPファイルを、直接編集することができるようになりました。

≺		test.csp
Ð	EXPLORER ····	≡ test.csp ×
	$\vee$ Workspace (Workspace)	1 <html></html>
$\cap$	✓ iris2021:NAKA	<pre>2 <head><title>test</title></head></pre>
$\succ$	> User	3 <body></body>
	= test mac	<pre>4 Hello world! at #(\$ZDT(\$H))#</pre>
P۵		5
0	✓ Iris2021:INAKA:csp	6
	✓ csp / naka	
	≡ test.csp	

# 6. まとめ

新しくなった InterSystems エクステンションシリーズによって、軽量で快適に動作する Visual Studio Code を、InterSystems IRIS サーバ上のコードエディタとしてお手軽にお使いいただけます。ぜひご活用くださいませ。

Happy Coding!

<u>#VSCode</u> <u>#InterSystems IRIS</u>

### ソースURL:

https://jp.community.intersystems.com/post/vscode-%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%A6%E3%81%BF%E3%82%88%E3%81%86-2021%E5%B9%B44%E6%9C%8820%E6%97%A5%E7%89%88