記事
Mihoko Iijima · 2021年4月22日 4m read

ソースプログラムを隠蔽化する方法

これは InterSystems FAQ サイト**の記事です。**

ルーチン（*.mac）の場合

ソースプログラムのコンパイル後に生成される *.obj
のみをエクスポート／インポートすることでソースの隠蔽化を実現できます。

コマンド実行例は、EX1Sample.mac と EX2Sample.mac のコンパイルで生成される EX1Sample.obj と EX2Sample.obj をエクスポート対象に指定し、第2引数のファイルにエクスポートしています。

別ネームスペースに移動したあと、エクスポートした XML ファイルを利用してインポートを実行しています。

```
USER>do $system.OBJ.Export("EX1Sample.obj,EX2Sample.obj","/opt/app/routine.xml")

XML????????? 04/22/2021 18:18:32
??????????????????: EX1Sample.obj
??????????????????: EX2Sample.obj
???????????????????

USER>zn "test"  // ?????????

TEST>do $system.OBJ.Load("/opt/app/routine.xml")

????? 04/22/2021 18:18:51
???? /opt/app/routine.xml ? xml ???????
??????????????????: EX1Sample
??????????????????: EX2Sample
??????????????????

TEST>
```

クラス（*.cls）の場合

クラスの場合は、XMLで *.cls をエクスポート／インポートしたあとに、サーバで
MakeClassDeployed() を実行します。

ただし、比較的新しいバージョンでは MakeClassDeployed() 実行後、ソースファイル（*.cls）は配置モードに設定されるのみで（編集はできなくなります）参照のみ行える仕様になっています。

参照も不可にしたい場合は、MakeClassDeployed() 実行後、クラスの Hidden
プロパティを設定します（プロパティの属性 Hidden=True に設定）。

コマンド実行例は以下の通りです。

```
USER>do $system.OBJ.Export("GPS.REST.cls,GPS.DriveData.cls","/opt/app/test.xml")

XML????????? 04/22/2021 18:05:13
??????????: GPS.DriveData
??????????: GPS.REST
???????????????

USER>zn "test"  // test??????????

TEST>do $system.OBJ.Load("/opt/app/test.xml","ck")

????? 04/22/2021 18:07:21
???? /opt/app/test.xml ? xml ???????
??????????: GPS.DriveData
??????????: GPS.REST
, 2 ??????????,  ????????????
?????????? GPS.DriveData
?????????? GPS.REST
?????????? GPS.DriveData
?????????? GPS.REST.1
?????????? GPS.DriveData.1
?????????????

TEST>do $system.OBJ.MakeClassDeployed("GPS.DriveData")

TEST>
```

CSP（*.csp）の場合

CSPファイルについては、*.cspをコピーし、配置先の CSP フォルダに貼付ます。

サーバでコンパイル後、CSPの設定で自動コンパイル OFF にしたあと、*.csp の削除と MakeClassDeployed() の実行を行います

実行例は以下の通りです。

１）CSPファイルのコピー後、サーバの CSP ディレクトリに貼付＆コンパイル

```
TEST>do $SYSTEM.CSP.LoadPageDir("/csp/test")
```

２）ウェブアプリケーションパスの設定で「自動コンパイル」をいいえに設定

【バージョン2013.1以降】 [管理ポータル] > [システム管理] > [セキュリティ] > [アプリケーション] > [ウェブ・アプリケーション] > 該当するアプリケーション名のリンク

【バージョン201.1～バージョン2012.2】 [管理ポータル] > [システム管理] > [セキュリティ] > [アプリケーション] > [ウェブ・アプリケーション] > 該当するアプリケーション名の[編集]

【バージョン2010.2以前】 [システム管理ポータル] > [システム] > [セキュリティ管理] > [CSPアプリケーション] > 該当するアプリケーション名の[編集]

３）MakeClassDeployed() の実行

※cspsample.csp をコピーした場合の例

```
TEST>do $system.OBJ.MakeClassDeployed("csp.cspsample")
```

#CSP #デプロイ #ヒントとコツ #Caché #Ensemble #HealthShare #InterSystems IRIS #InterSystems IRIS for Health

---

ソースURL:
https://jp.community.intersystems.com/post/%E3%82%BD%E3%83%BC%E3%82%B9%E3%83%97%E3%83%AD%E3%82%B0%E3%83%A9%E3%83%A0%E3%82%92%E9%9A%A0%E8%94%BD%E5%8C%96%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95