

記事

[Toshihiko Minamoto](#) · 2021年10月19日 9m read

%Statusを使ったデバッグのヒント

はじめに

ObjectScriptで複雑な問題を解決している場合、おそらく%Status値を使用したコードがたくさんあることでしょう。オブジェクトの観点(%Save、%OpenIdなど)から永続クラスを操作したことがある場合は、ほぼ確実にその状況に遭遇したことがあるでしょう。

%StatusはInterSystemsのプラットフォームでローカライズ可能なエラーメッセージのラッパーを提供します。OKステータス(\$\$\$OK)は1に等しいだけであるのに対し、不良ステータス(\$\$ERROR(errorcode,arguments...))は0、スペース、エラーに関する構造化情報を含む\$ListBuildリストとして表されます。

[\\$System.Status \(クラスリファレンスを参照\)](#)

は、%Status値を操作するための便利なAPIをいくつか提供しています。クラスリファレンスを役立てられるので、ここでは繰り返しません。

このトピックに関する有用な記事/質問もほかにいくつかあります(最後のリンクをご覧ください)。この記事では、コーディングのベストプラクティスではなく、いくつかのデバッグのコツや手法に焦点を当てています(ベストプラクティスについては、最後のリンクをご覧ください)。

お題のコード例

注意: このようなコードは絶対に書かないようにしてください！

常にステータスをチェックし、それを返すか、例外としてスロー(\$\$\$ThrowStatus(someErrorStatus)など)すれば、デバッグがはるかに簡単になります。

```
Class DC.Demo.MaskedErrorStatus Extends %Persistent
{
    Property Answer As %TinyInt;

    ClassMethod Run() As %Status
    {
        Set instance = ..%New()
        Set instance.Answer = 9000
        Do instance.%Save()

        Set instance = ..%OpenId(1,,.sc)
        Set instance.Answer = 42
        Do instance.%Save()

        Quit $$$OK
    }
}
```

ターミナルから実行すると、例外がスローされます。何かが明らかにうまく行っていません。

```
USER>d ##class(DC.Demo.MaskedErrorStatus).Run()
```

```
Set instance.Answer = 42
^
<INVALID OREF>zRun+5^DC.Demo.MaskedErrorStatus.1
```

%Statusのデバッグのコツ#1: \$System.OBJ.DisplayError()

常に \$System.OBJ.DisplayError() を実行して、作成された最後のエラーステータスを出力できます。これが機能するのは、(\$System.Status.Error 経由で) エラーステータスが作成されるたびに、変数 %objlasterror にそのステータスが設定されるためです。また、同様に %objlasterror を zwrite することもできます。上記の場合は、次のようになります。

```
USER 2d1>d $system.OBJ.DisplayError()
ERROR #5809: Object to Load not found, class 'DC.Demo.MaskedErrorStatus', ID '1'
```

%Statusのデバッグのコツ#2: スタックトレース

%Status ごとに、エラーが作成された場所のスタックトレースがあります。ステータスを zwrite することで、そのトレースを閲覧することができます。

```
USER 2d1>zw %objlasterror
%objlasterror="0"_$lb($lb(5809,"DC.Demo.MaskedErrorStatus","1",,,,,,$lb(",USER",$lb(
"e^%LoadData+18^DC.Demo.MaskedErrorStatus.1^1","e^%Open+16^%Library.Persistent.1^1","
e^%OpenId+1^%Library.Persistent.1^1","e^zRun+4^DC.Demo.MaskedErrorStatus.1^1","d^^^0"
))))/* ERROR #5809: Object to Load not found, class 'DC.Demo.MaskedErrorStatus', ID '
1' */
```

ステータスごとに、より分かりやすいテキスト (\$System.OBJ.DisplayError() または \$System.Status.GetErrorText(someStatus)) でスタックトレースを確認したいですか？ これは、^%oddENV("callererrorinfo",\$namespace)=1 または 2 に設定することで実現できます。以下に、その効果を示します。

```
USER>set ^%oddENV("callererrorinfo",$namespace)=1
USER>d $system.OBJ.DisplayError()
ERROR #5809: Object to Load not found, class 'DC.Demo.MaskedErrorStatus', ID '1' [%Lo
adData+18^DC.Demo.MaskedErrorStatus.1:USER]
USER>set ^%oddENV("callererrorinfo",$namespace)=2
USER>d $system.OBJ.DisplayError()
ERROR #5809: Object to Load not found, class 'DC.Demo.MaskedErrorStatus', ID '1' [e^%
LoadData+18^DC.Demo.MaskedErrorStatus.1^1 e^%Open+16^%Library.Persistent.1^1 e^%OpenI
d+1^%Library.Persistent.1^1 e^zRun+4^DC.Demo.MaskedErrorStatus.1^1 d^^^0:USER]
USER>k ^%oddENV("callererrorinfo",$namespace)
USER>d $system.OBJ.DisplayError()
ERROR #5809: Object to Load not found, class 'DC.Demo.MaskedErrorStatus', ID '1'
```

これは実際には、開発環境のみで使用するのが適切であることに注意してください。ユーザーにはコードの内部を見られたくないはずです。(ユーザーに直接 %Status 値を表示することを避けて、よりユーザーフレンドリーなアプリケーション固有のエラーメッセージを表示するのが最善ですが、これについては、別のトピックとしましょう。)

%Statusのデバッグのコツ#3: 極上の zbreak

この辺からトリッキーになってきます。このコードスニペットの場合では、根本的な原因は、前述のコードスニペットの %Save() から %Status が確認されていないことにあります。何がうまく行かなかったのかを見つけるのが非常に難しい、はるかに複雑な例を想像するのは簡単です。特に、プラットフォームコードのずっと後の方で発生

%Statusを使ったデバッグのヒント

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

しているエラーであれば、尚更です。これに対処するためにインタラクティブデバッガに飛びつく以外で私が好んで使用してる方法は、ターミナルで非常に極上の**zbreak**コマンドを使用することです。

```
USER>zbreak *%objlasterror:"N": "$d(%objlasterror)#2": "set ^mtemptl($i(^mtemptl))=%objlasterror"
```

これはどういう意味でしょうか？

zbreak <%objlasterrorが変更される時点>:<デバッガでは何もしない>:<%objlasterrorが定義されており値がある限り（定義済みから未定義にならない場合など）>:<整数がサブスクリプトのジャーナルされていないグローバル（mtempから始めることで、トランザクション中に%Statusが作成され、ログを見るより先にロールバックされてしまうの防ぐため。また、コミットされたコードやデータベースで誰かがこれを見つけた時に私に連絡できるように、グローバルの一部に私のイニシャルを使っています）で、次のサブスクリプトのグローバルにエラーステータスをセットする処理を実行>

zbreak に関する補足: 現在定義されているブレイクポイント/ウォッチポイントは、引数を指定せずに 'zbreak' を実行することで確認できます。また、ブレイクポイントの使用が終わったら、break "off" などを実行して、これらのブレイクポイントをオフにできますし、する必要があります。以下に例を示します。

```
USER>zbreak
BREAK:
  No breakpoints
%objlasterror F:E S:0 C:"$d(%objlasterror)#2" E:"set ^mtemptl($i(^mtemptl))=%objlasterror"
USER>break "off"
USER>zbreak
BREAK:
  No breakpoints
  No watchpoints
```

では、問題のあるメソッドがウォッチポイントを設定して実行されるとどうなるのでしょうか？

```
USER>zbreak *%objlasterror:"N": "$d(%objlasterror)#2": "set ^mtemptl($i(^mtemptl))=%objlasterror"
```

```
USER>d ##class(DC.Demo.MaskedErrorStatus).Run()
```

```
Set instance.Answer = 42
^
<INVALID OREF>zRun+5^DC.Demo.MaskedErrorStatus.1
```

```
USER 2d1>zw ^mtemptl
^mtemptl=6
^mtemptl(1)="0 "_$lb($lb(7203,9000,127,,,,,,,$lb(",USER",$lb("e^zAnswerIsValid+1^DC.Demo.MaskedErrorStatus.1^1","e^%ValidateObject+3^DC.Demo.MaskedErrorStatus.1^4","e^%SerializeObject+3^%Library.Persistent.1^1","e^%Save+4^%Library.Persistent.1^2","d^zRun+3^DC.Demo.MaskedErrorStatus.1^1","d^^^0"))))/* ERROR #7203: Datatype value '9000' greater than MAXVAL allowed of 127 */
^mtemptl(2)="0 "_$lb($lb(7203,9000,127,,,,,,,$lb(",USER",$lb("e^zAnswerIsValid+1^DC.Demo.MaskedErrorStatus.1^1","e^%ValidateObject+3^DC.Demo.MaskedErrorStatus.1^4","e^%SerializeObject+3^%Library.Persistent.1^1","e^%Save+4^%Library.Persistent.1^2","d^zRun+3^DC.Demo.MaskedErrorStatus.1^1","d^^^0")),0 "_$lb($lb(5802,"DC.Demo.MaskedErrorStatus:Answer",9000,,,,,,,$lb(",USER",$lb("e^EmbedErr+1^%occSystem^1"))))/* ERROR #7203: Datatype value '9000' greater than MAXVAL allowed of 127- > ERROR #5802: Datatype validation failed on property 'DC.Demo.MaskedErrorStatus:Answer', with value equal to 0 "9000" */
```

```
^mtemptl(3)="0 "_$lb($lb(7203,9000,127,,,,,,,$lb("zAnswerIsValid+1^DC.Demo.MaskedErrorStatus.1","USER",$lb("e^zAnswerIsValid+1^DC.Demo.MaskedErrorStatus.1^1","e^%ValidateObject+3^DC.Demo.MaskedErrorStatus.1^4","e^%SerializeObject+3^%Library.Persistent.1^1","e^%Save+4^%Library.Persistent.1^2","d^zRun+3^DC.Demo.MaskedErrorStatus.1^1","d^^^0"))))/ * ERROR #7203: Datatype value '9000' greater than MAXVAL allowed of 127 */
^mtemptl(4)="0 "_$lb($lb(5802,"DC.Demo.MaskedErrorStatus:Answer",9000,,,,,,,$lb("EmbedErr+1^%occSystem","USER",$lb("e^EmbedErr+1^%occSystem^1"))))/ * ERROR #5802: Datatype validation failed on property 'DC.Demo.MaskedErrorStatus:Answer', with value equal to '9000' */
^mtemptl(5)="0 "_$lb($lb(7203,9000,127,,,,,,,$lb("zAnswerIsValid+1^DC.Demo.MaskedErrorStatus.1","USER",$lb("e^zAnswerIsValid+1^DC.Demo.MaskedErrorStatus.1^1","e^%ValidateObject+3^DC.Demo.MaskedErrorStatus.1^4","e^%SerializeObject+3^%Library.Persistent.1^1","e^%Save+4^%Library.Persistent.1^2","d^zRun+3^DC.Demo.MaskedErrorStatus.1^1","d^^^0"))),"0 "_$lb($lb(5802,"DC.Demo.MaskedErrorStatus:Answer",9000,,,,,,,$lb("EmbedErr+1^%occSystem","USER",$lb("e^EmbedErr+1^%occSystem^1")))))/ * ERROR #7203: Datatype value '9000' greater than MAXVAL allowed of 127- > ERROR #5802: Datatype validation failed on property 'DC.Demo.MaskedErrorStatus:Answer', with value equal to '9000' */
^mtemptl(6)="0 "_$lb($lb(5809,"DC.Demo.MaskedErrorStatus","1",,,,,,$lb(",USER",$lb("e^LoadData+18^DC.Demo.MaskedErrorStatus.1^1","e^%Open+16^%Library.Persistent.1^1","e^%OpenId+1^%Library.Persistent.1^1","e^zRun+4^DC.Demo.MaskedErrorStatus.1^1","d^^^0"))))/ * ERROR #5809: Object to Load not found, class 'DC.Demo.MaskedErrorStatus', ID '1' */
```

多少ノイズがありますが、重要な問題がすぐに現れます。

```
/* ERROR #7203: Datatype value '9000' greater than MAXVAL allowed of 127 */
```

%TinyInt を使用しないことを想定すべきでした！

(また、より重要なことは、呼び出すメソッドが返す%Status値を必ず確認する必要があります。)

関連資料

[私が気に入っている、エラー処理とレポートのコーディングパターン](#)

[Caché ObjectScriptメソッドの%Statusとその他の戻り値](#)

[%objlasterror!について](#)

[\\$\\$EnvCallerErrorInfoGetの設定方法](#)

[ObjectScriptによるエラー処理スニペット](#)

[ZBREAKコマンド](#)

[#エラーハンドリング](#) [#ObjectScript](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

ソースURL:

<https://jp.community.intersystems.com/post/status%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%9F%E3%83%87%E3%83%90%E3%83%83%E3%82%B0%E3%81%AE%E3%83%92%E3%83%B3%E3%83%88>