

# AutoLISPからEXCELシートの読み書き

「AutoCAD のデータをEXCELシートに表示する。そして、EXCELシート上でデータを編集してAutoCAD に反映する。」という目的への第一歩として、AutoLISP からEXCELシートを読み書きする方法を説明します。

## ■ ActiveX 経由で Excel を操作する

WindowS版のAutoCAD(AutoLISP)はActive X をサポート関数を備えていて、Microsoft Excel アプリケーションに接続することが可能で、エクセルオブジェクトのメソッド、プロパティ、コレクションをコントロールすることができるということです。

平たく言うと、ExcelVBAで行えることと同様のことが、AutoLISPでもできるのだと思っています。

まずは、手始めとして、簡単なプログラムを作ってみたいと思います。

## ■ サンプルプログラム

これは、「指定のワークシートを読みだして、指定のセルの値を読み取り、指定のセルにその値を書き込む。」というコマンドです。

```
(Defun C:TEST ()  
;  
(vl-load-com)  
;  
(setq filepath "C:\\Users\\YY4073481\\Documents\\YYBook1.xlsx")  
(setq sheetname "Sheet1")  
(setq i 1 j 1)  
(setq k 10 l 1)  
;  
(setq Obj_ExcelApplication (vlax-get-or-create-object "Excel.Application"))  
(vlax-put-property Obj_ExcelApplication 'Visible :vlax-true)  
(setq Col_Workbooks (vlax-get-property Obj_ExcelApplication 'Workbooks))  
(setq Obj_Workbook (vlax-invoke-method Col_Workbooks 'Open filepath))  
(setq Col_Worksheets (vlax-get-property Obj_Workbook 'Worksheets))  
(setq Obj_Sheet (vlax-get-property Col_Worksheets 'Item sheetname))  
(vlax-invoke-method Obj_Sheet 'Activate)  
;  
(setq Col_Range (vlax-get-property Obj_Sheet 'Cells))  
(setq Obj_Range (vlax-get-property Col_Range 'Item i j))  
(setq Obj_Range (vlax-variant-value Obj_Range))  
(setq Val_var (vlax-get-property Obj_Range 'Value))  
(setq Val (vlax-variant-value Val_var))  
;  
(setq Obj_Range (vlax-get-property Col_Range 'Item k l))  
(setq Obj_Range (vlax-variant-value Obj_Range))  
(vlax-put-property Obj_Range 'Value2 Val_var)  
;  
(vlax-release-object Obj_Range)  
(vlax-release-object Obj_Sheet)  
(vlax-release-object Col_Worksheets)  
(vlax-release-object Obj_Workbook)  
(vlax-release-object Col_Workbooks)  
(vlax-release-object Obj_ExcelApplication)  
;  
(princ)  
)
```



実際の環境に合わせて変更してください。

## ■ サンプルプログラムの説明（1／4）

緑色太字は、エクセル VLA オブジェクト（コレクション）変数です。

|                             |                    |
|-----------------------------|--------------------|
| <b>Obj_ExcelApplication</b> | エクセルアプリケーションオブジェクト |
| <b>Col_Workbooks</b>        | ワークブックコレクション       |
| <b>Obj_Workbook</b>         | ワークブックオブジェクト       |
| <b>Col_Worksheets</b>       | ワークシートコレクション       |
| <b>Obj_Sheet</b>            | シートオブジェクト          |
| <b>Col_Range</b>            | レンジコレクション（オブジェクト？） |
| <b>Obj_Range</b>            | レンジオブジェクト          |

紫色太字は、その他の変数です。

|                  |                              |
|------------------|------------------------------|
| <b>filepath</b>  | 呼び出すエクセルワークブックのフルパス          |
| <b>sheetname</b> | 参照するシート名                     |
| <b>i,j</b>       | 値を読みだすセルのローアドレスとコラムアドレス      |
| <b>k,l</b>       | 読みだした値を書き込むセルのローアドレスとコラムアドレス |
| <b>Val_var</b>   | 読みだした値（バリエーション型）             |
| <b>Val</b>       | 非バリエーション型に変換した値              |



実際に合わせて設定してください。

青色太字は Active X 関数です。

|                                  |                                                 |
|----------------------------------|-------------------------------------------------|
| <b>vl-load-com</b>               | 拡張機能である Visual LISP、Active X を使用可能にします。         |
| <b>vlax-get-or-create-object</b> | アプリケーションオブジェクトの実行中インスタンスを返すか、無い場合はインスタンスを作成します。 |
| <b>vlax-get-property</b>         | VLA オブジェクトのプロパティを取得します。                         |
| <b>vlax-put-property</b>         | VLA オブジェクトのプロパティを設定します。                         |
| <b>vlax-invoke-method</b>        | メソッドを呼び出します。                                    |
| <b>vlax-variant-value</b>        | バリエーション型変数の値を非バリエーション型に変換して返します。                |
| <b>vlax-release-object</b>       | VLA オブジェクトを開放します。                               |

赤色太字は関数と組み合わせて、オブジェクト、プロパティ、メソッドなどを指定します。

|                            |                                                 |
|----------------------------|-------------------------------------------------|
| <b>"Excel.Application"</b> | エクセルアプリケーション                                    |
| <b>'Workbooks</b>          | ワークブックコレクション                                    |
| <b>'Worksheets</b>         | ワークシートコレクション                                    |
| <b>'Item</b>               | コレクション内オブジェクト（引数にて特定）                           |
| <b>'Cells</b>              | Range コレクション（オブジェクト？）※ シート内のすべてのセルをイメージすればよい？   |
| <b>'Value</b>              | Range.Value プロパティ                               |
| <b>'Value2</b>             | Range.Value プロパティ（put のときは、Value2 でないとうまくいかない。） |
| <b>'Visible</b>            | Visible メソッド                                    |
| <b>'Open</b>               | Open メソッド                                       |
| <b>'Activate</b>           | Activate メソッド                                   |

## ■ サンプルプログラムの説明（2 / 4）

### 3. (vl-load-com)

3. 拡張機能である Visual LISP、Active X を使用可能にします。

5. (setq filepath "C:¥¥Users¥¥4073481¥¥Documents¥¥Book1.xlsx")

6. (setq sheetname "Sheet1")

7. (setq i 1 j 1)

8. (setq k 10 l 1)

変数に値をセットします。

5. filepath = “C:/Users/4073481/Documents/Book1.xlsx”

6. sheetname = "Sheet1“

7. i = 1、j = 1

8. k = 10、l = 1

10. (setq Obj\_ExcelApplication (vlax-get-or-create-object "Excel.Application"))

11. (vlax-put-property Obj\_ExcelApplication 'Visible :vlax-true)

12. (setq Col\_Workbooks (vlax-get-property Obj\_ExcelApplication 'Workbooks))

13. (setq Obj\_Workbook (vlax-invoke-method Col\_Workbooks 'Open filepath))

14. (setq Col\_Worksheets (vlax-get-property Obj\_Workbook 'Worksheets))

15. (setq Obj\_Sheet (vlax-get-property Col\_Worksheets 'Item sheetname))

16. (vlax-invoke-method Obj\_Sheet 'Activate)

10. エクセルアプリケーションを起動し、

11. 可視化します。

12. ワークブックコレクションから

13. 「Bokk1.xlsx」をオープンします。

14. ワークシートコレクションの中から

15. “Sheet1”を

16. アクティブにします。

## ■ サンプルプログラムの説明（3 / 4）

18. (setq Col\_Range (vlax-get-property Obj\_Sheet 'Cells))
19. (setq Obj\_Range (vlax-get-property Col\_Range 'Item i j))
20. (setq Obj\_Range (vlax-variant-value Obj\_Range))
21. (setq Val\_var (vlax-get-property Obj\_Range 'Value))
22. (setq Val (vlax-variant-value Val\_var))

18. シート内のセルから **注1)**
19. セル（1, 1）Rangeオブジェクトを取得し
20. 非バリエーション型に変換
21. セルの値を取得、
22. 非バリエーション型に変換します。

**注1)** Rangeにコレクションは無いとのことですが、範囲が決まっていまいRangeなので、あえて「Col\_Range」というオブジェクト名にしています。

24. (setq Obj\_Range (vlax-get-property Col\_Range 'Item k l))
25. (setq Obj\_Range (vlax-variant-value Obj\_Range))
26. (vlax-put-property Obj\_Range 'Value2 Val\_var)

24. セル（10, 1）Rangeオブジェクトを取得し
25. 非バリエーション型に変換
26. 「21.」にて読み込んだ値を設定します。 **注2)**

**注2)** putするのは、Val でもよさそうです。プロパティ名は、'Value ではNGでした。（理由不明）

28. (vlax-release-object Obj\_Range)
29. (vlax-release-object Obj\_Sheet)
30. (vlax-release-object Col\_Worksheets)
31. (vlax-release-object Obj\_Workbook)
32. (vlax-release-object Col\_Workbooks)
33. (vlax-release-object Obj\_ExcelApplication)

28.～33. 取得したVLAオブジェクトを開放します。

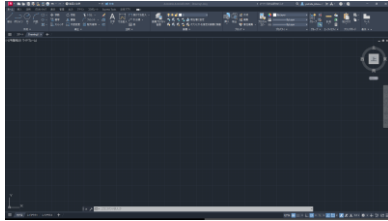
35. (princ)

35. 戻り値不要のため (princ) で終了します。

## ■ サンプルプログラムの説明（4 / 4）

いろいろと、誤りや説明不足があるとは思いますが、お許してください。  
以上で、説明を終わります。

一連の動画です。



クリック  
動画が再生されます

- ① サンプルプログラム「TEST」を メモ帳からコピーして コマンドラインに流し込みます。
- ② コマンド「TEST」を実行すると、「Book1.xlsx」が開きます。
- ③ Cells(1, 1) の値""が読み込まれて、Cells(10, 1) に書き込まれたはずですが 確認することはできません。
- ③ Cells(1, 1) に"文字"と書き込み、「TEST」を実行します。
- ④ Cells(1, 1) の値"文字"が読み込まれて、Cells(10, 1) に書き込まれたことが確認できます。
- ⑤ Cells(1, 1) に"123456"と書き込み、「TEST」を実行します。
- ⑥ Cells(1, 1) の値"123456"が読み込まれて、Cells(10, 1) に書き込まれたことが確認できます。

以上のように、Active X 関数を用いて エクセルのオブジェクトをコントロールすれば、AutoLISP にて取得したCAD情報を エクセルシートにて記述したり、そのデータをエクセル上で編集してから、AutoCAD に反映するなどのカスタマイズコマンドの作成が可能になります。

以降では、その具体例について記述していきたいと思います。