

# AL7\_Transpose2D

## PRESENTATION

### I. Introduction

L'UDF (User Defined Function) **AL7\_Transpose2D** offre une solution avancée et performante pour transposer des matrices ou des plages de cellules en Excel, surpassant Application.Transpose en termes de robustesse et de flexibilité. Cette fonction se décline en deux versions :

- **Transpose2D (alias Trans2D) :**  
Principalement destinée à une utilisation dans VBA.
- **TransRange (alias TransRng) :**  
Certaines options ne sont pas accessibles afin d'être utilisée plus facilement dans les cellules. Si besoin utilisez la fonction Transpose2D (*Trans2D*).

Bien sûr les deux fonctions peuvent être utilisées dans les cellules ou dans VBA, la différence est que Transpose2D propose plus de fonctions que TransRange.

**Tout en gardant les avantages de Transpose2D, vous pouvez demander à garder les mêmes caractéristiques/comportement de numérotation des transpositions que les fonctions Transpose d'Excel pour cela mettez l'option LikeExcel = True (-1).**

### Fonctions supplémentaires:

**TRANSPOSITION PARTIELLE** du tableau en indiquant le début et fin de chaque dimension.

**REDIMENSIONNEMENT INDÉPENDANT** de chaque dimension.

... voir tableau comparatif ci-après.

## A. Comparatif

### 1. Transpose2D, Application.Transpose, et WorksheetFunction.Transpose

	Transpose2D	Application.Transpose	WorksheetFunction.Transpose
<b>Taille maximale gérée</b>	Illimitée (optimisé). Limite: Capacité matériel	Dépend de la version d'Excel	<b>65 536 éléments max</b> (erreur au-delà)
<b>Renvoie toujours un tableau.</b> Cas de tableau T(1,1) ou T(1 to 1)	<b>Oui.</b> Sauf si l'option LikeExcel = False	<b>Non</b>	<b>Non</b>
<b>Convertit une simple chaîne ou une valeur en tableau</b>	<b>Oui</b> , 99 → T(1,1,) = 99 "Foo" → T(1,1) = "Foo"	<b>Non</b>	<b>Non</b>
<b>Si 2ème dimension = 1, conversion en tableau à 1 dimension.</b> Ex.: T(5,1) → T(5)	<b>A la demande</b> (Option: LikeExcel = True).	<b>Oui, toujours</b>	<b>Oui, toujours</b>
<b>Bases des 2 dimensions indépendantes et paramétrables</b>	<b>OUI</b> , Utilisez Transpose2D	<b>NON</b> , toujours base = 1 → LBound de T,1 et T,2 = 1	<b>NON</b> , toujours base = 1 → LBound de T,1 et T,2 = 1
<b>Gestion des erreurs</b>	<b>Oui</b> , avec messages clairs	Non, erreurs fréquentes	Non, erreurs fréquentes
<b>Conversion automatique des types</b>	<b>Oui</b>	<b>Oui</b> , mais parfois instable	<b>Non</b> (erreurs avec les nombres longs)
<b>Extraction et transposition partielle</b>	<b>Oui</b> , sur chacune des 2 dimensions.	<b>Non</b>	<b>Non</b>
<b>Utilisation avec des plages nommées</b>	<b>Oui</b>	<b>Oui</b>	<b>Oui</b>
<b>Utilisation sur plusieurs feuilles de classeurs ouvert ou fermés</b>	<b>Oui</b>	<b>Non</b>	<b>Non</b>
<b>Compatibilité avec les tableaux dynamiques</b>	<b>Oui</b>	<b>Oui</b> (Excel 365/2019)	<b>Non</b>
<b>Performance sur grandes matrices</b>	<b>Optimisée</b> (traitement rapide)	<b>Bonne</b> mais peut ralentir	<b>Problématique</b> au-delà de 65 536 éléments
<b>Utilisation en VBA</b>	<b>Oui</b> , gestion complète des paramètres	<b>Oui</b> , mais moins flexible	<b>Oui</b> , mais nécessite un tableau explicite
<b>Utilisation en cellules Excel</b>	<b>Oui</b>	<b>Oui</b>	<b>Oui</b> (mais limité en fonctionnalités)
<b>Conserver les bases de chaque dimension</b>	<b>Oui</b>	<b>Non</b> (Retourne toujours les tableaux en base 1)	<b>Non</b> (Retourne toujours les tableaux en base 1)
<b>Modifier chacune des bases de chaque dimension</b>	<b>Oui.</b> Utilisez Transpose2D)	<b>Non</b> (Retourne toujours les tableaux en base 1)	<b>Non</b>

## B. Cas pratiques et recommandations

- ✓ **Traitement de grands ensembles de données** : Plus rapide et sans limitation.
- ✓ **Transposition partielle d'un tableau** sur une ou 2 dimensions.
- ✓ **Utilisation avec des nombres et du texte** : Contrairement à Application.Transpose, aucune erreur de conversion.
- ✓ **Compatible avec les tableaux dynamiques** : Intégration fluide avec les nouvelles fonctionnalités Excel.
- ✓ **Utilisation avec des plages nommées et des classeurs fermés**.

## C. Deux types de fonctions pour des usages spécifiques

### 1. Transpose2D (alias Trans2D) – Orienté VBA

- ✓ **Utilisation principale** :  
Fonction optimisée pour une utilisation en VBA car **elle comporte plus de possibilités que TransRange**.
- ✓ **Pourquoi l'utiliser ?**  
Gère mieux les tableaux dynamiques et évite les erreurs classiques d'Application.Transpose.
- ✓ **Paramètres** :  
`TRANSPOSE2D(ToTranspose, _  
[LikeExcel], _  
[StartDim1], _  
[EndDim1], _  
[StartDim2], _  
[EndDim2], _  
{KeepNumDim1}, _  
[KeepNumDim2], _  
[BaseOut1], _  
[BaseOut2])`

#### Exemples en VBA :

```
Dim Result As Variant
' Transposition simple
Result = Trans2D(MyArray)
' Ou transposition d'une partie d'un tableau : de 1 à 5 et de 2 à 4
Result = Transpose2D(MyArray, LikeExcel:=False; StartDim1:=1; EndDim1:=5; StartDim2:=2; EndDim2:=4)
```

### 2. TransRange (alias TransRng) – Orienté cellules Excel

- ✓ **Utilisation principale** :  
Fonction destinée à être utilisée directement dans les feuilles Excel.
- ✓ **Pourquoi l'utiliser ?**  
Plus simple car moins de paramètres, elle suffit souvent à transposer en tout ou partie une plage. Permet une intégration fluide dans les formules Excel sans nécessiter VBA.
- ✓ **Paramètres** :  
`TransRange(Transpose_Range, _  
[StartDim1], _  
[EndDim1], _  
[StartDim2], _  
[EndDim2])`

#### Exemple en cellule Excel :

```
' Transposition simple
=TransRange(Plage de cellules)
' Transposition partielle de la plage A1:D3 de 1 à 3 et de 1 à 4
=TransRng(A1:D3, 1, 3, 1, 4)
```



# Guide d'utilisation rapide

Dans ce guide nous vous présentons les différents paramètres que vous pouvez utiliser.

Si besoin, une succincte explication de leurs utilisations vous sera donnée.

Les paramètres entre [ ... ] sont optionnels.

A paramètres identiques, les fonctions Transpose2D (alias Trans2D) et TransRange (alias TransRng) sont interchangeables. Donc vous obtiendrez les mêmes résultats.

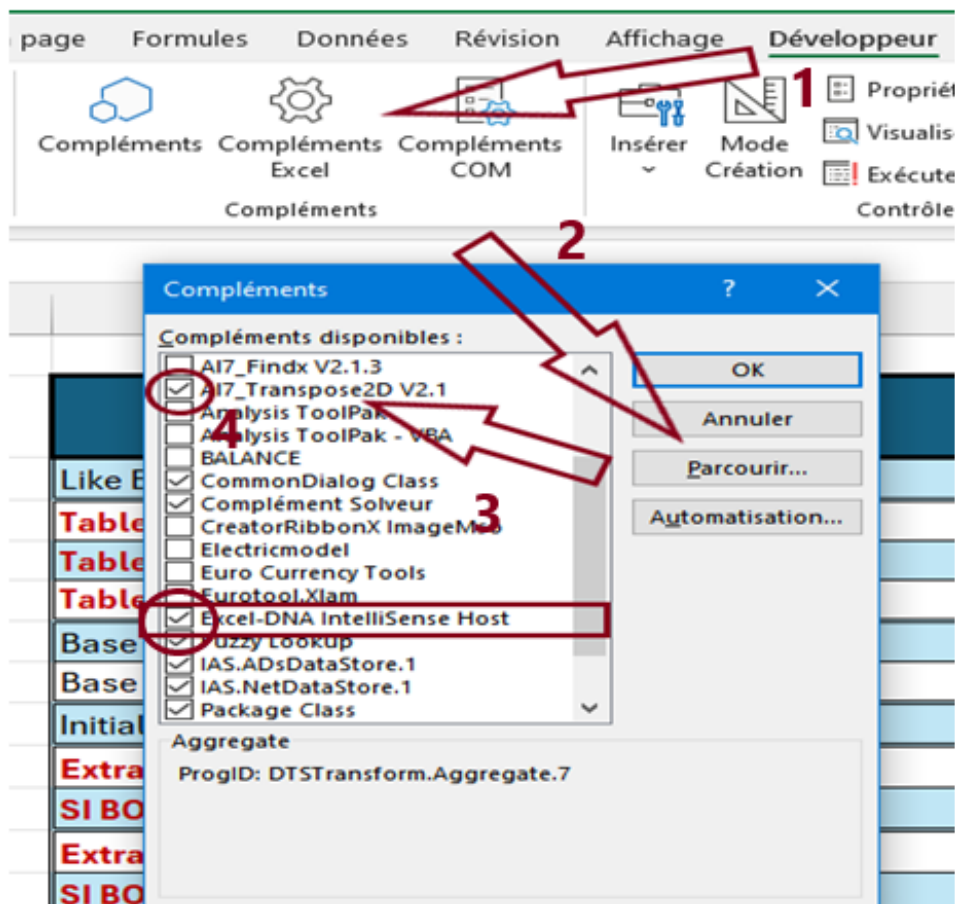
Bien que les exemples de Transpose2D indiquent des tableaux vous pouvez fournir une plage d'adresse Excel. Ex. : `Range("A1:F100")`.

## I. Installation et configuration

### 1. Installation sur disque dur

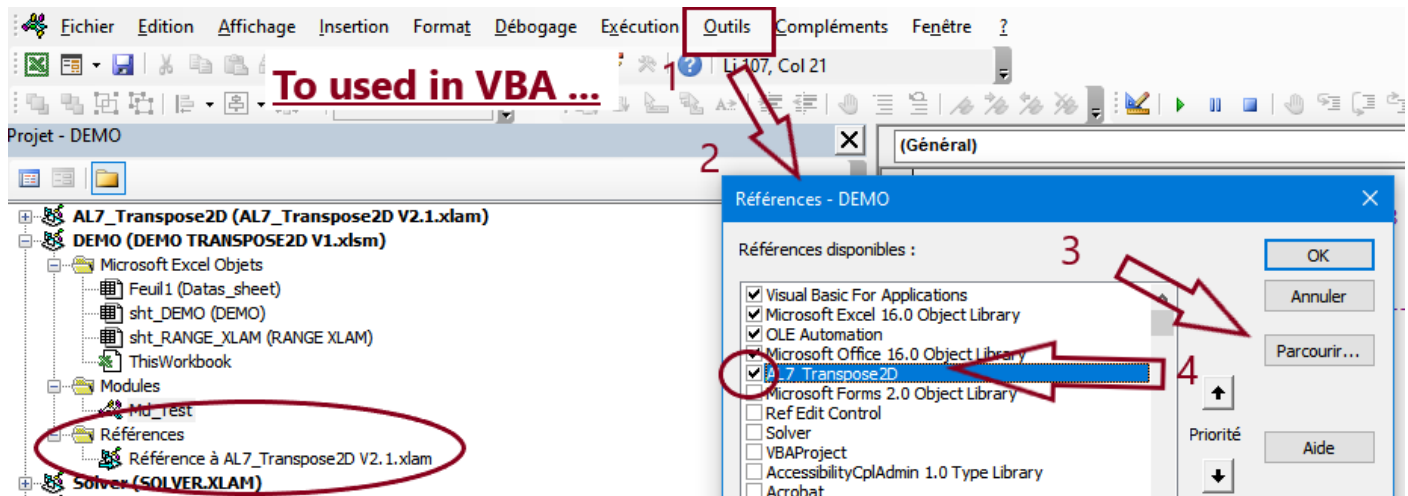
- Téléchargez *AL7\_Transpose2D.zip*
- Décompresser/Extraire le fichier pour avoir l'ensemble des fichiers.
- Placez le fichier *AL7\_Transpose2D.xlam* dans un dossier accessible  
**En général:** C:\Users\VotreNom\AppData\Roaming\Microsoft\AddIns
- Placez le fichier *ExcelDna.IntelliSense64.xll* dans le même répertoire.
- Ouvrez Excel et accédez à **Fichier > Options > Compléments**.
- Dans la section **Gérer**, sélectionnez **Compléments Excel** et cliquez sur **Atteindre**.
- Cliquez sur **Parcourir...** et sélectionnez *AL7\_Transpose2D.xlam*.
- Répéter les mêmes opérations pour le fichier *ExcelDna.IntelliSense64.xll*
- Validez et assurez-vous que les 2 add-in sont activés.

N'oubliez pas de référencer *ExcelDna.IntelliSense64* afin d'avoir l'aide à la saisie des paramètres des fonctions.



## 2. Référencement dans VBA

1. Ouvrez **VBA (ALT + F11)**.
2. Accédez à **Outils > Références**.
3. Cochez **AL7\_Transpose2D** pour permettre son utilisation dans votre programme VBA.



## II. Utilisation de la fonction Transpose2D et TransRange

### 1. Transpose2D: Syntaxe VBA

```
Result_arr = Transpose2D(Tableau_de_données, _  
                          [StartDim1], _  
                          [EndDim1], _  
                          [StartDim2], _  
                          [EndDim2], _  
                          [BaseOut1], _  
                          [BaseOut2])
```

Les paramètres entre [ ... ] sont optionnels.

- **Tableau\_de\_données** : Un tableau ou une plage de cellules
- **[StartDim1]** : Dans la 1<sup>er</sup> dimension du tableau, le numéro de début à transposer
- **[EndDim1]** : Dans la 1<sup>er</sup> dimension du tableau, le numéro de fin à transposer
- **[StartDim2]** : Dans la 2<sup>ème</sup> dimension du tableau, le numéro de début à transposer
- **[EndDim2]** : Dans la 2<sup>ème</sup> dimension du tableau, le numéro de fin à transposer
- **[BaseOut1]** : nombre servant de base à la 1<sup>er</sup> dimension du tableau.
- **[BaseOut2]** : nombre servant de base à la 2<sup>ème</sup> dimension du tableau.

### 2. Exemples:

#### Exemple en VBA :

```
Dim Result_arr As Variant
```

*'1) Transposition simple*

```
Result_arr = Trans2D(MyArray)
```

*'2) Transposition d'une partie d'un tableau : de 1 à 5 et de 2 à 4*

```
Result_arr = Transpose2D(MyArray; LikeExcel:=False; StartDim1:=1; EndDim1:=5; StartDim2:=2; EndDim2:=4)
```

*'3) Transposition en conservant en changeant la base de l'une ou des deux dimensions*

*' Ici on demande que le tableau transposé commence à 10 et à 100.*

```
Result_arr = Transpose2D(MyArray; BaseOut1:=10; BaseOut2:=100)
```

### 3. Syntaxe dans une cellule:

= TransRange (Plage\_de\_données, [StartDim1], [EndDim1], [StartDim2], [EndDim2])

**Exemple avec TransRange dans une cellule :**

*' Transposition simple pour obtenir la transposition*

= TransRange(A1:D3)

SOMME		=TRANSRNG(B4:E11)														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
									TRANSPOSE A SELECTED AREA							
									TRANSRANGE(B7:E14)							
2	ColonnE1	ColonnE2	ColonnE3	ColonnE4	ColonnE5	ColonnE6	ColonnE7		E11)	B5	B6	B7	B8	B9	B10	B11
3	A3	B3	C3	D3	E3	F3	G3		C4	C5	C6	C7	C8	C9	C10	C11
4	A4	B4	C4	D4	E4	F4	G4		D4	D5	D6	D7	D8	D9	D10	D11
5	A5	B5	C5	D5	E5	F5	G5		E4	E5	E6	E7	E8	E9	E10	E11
6	A6	B6	C6	D6	E6	F6	G6									
7	A7	B7	C7	D7	E7	F7	G7									
8	A8	B8	C8	D8	E8	F8	G8		IDEM FROM ANOTHER LOCAL SHEET							
9	A9	B9	C9	D9	E9	F9	G9		TRANSRANGE(Feuil11B7:E14)							
0	A10	B10	C10	D10	E10	F10	G10		B7	B8	B9	B10	B11	B12	B13	B14
1	A11	B11	C11	D11	E11	F11	G11		C7	C8	C9	C10	C11	C12	C13	C14
2	A12	B12	C12	D12	E12	F12	G12		D7	D8	D9	D10	D11	D12	D13	D14
3	A13	B13	C13	D13	E13	F13	G13		E7	E8	E9	E10	E11	E12	E13	E14

## III. Conclusion

Avec **AL7\_Transpose2D**, gagnez en efficacité et en fiabilité dans la manipulation de vos matrices Excel.

Conçu pour surpasser Transpose d'Excel, cet outil est indispensable pour tout utilisateur manipulant des données complexes.

Téléchargez-le, activez et profitez d'une transposition optimisée!