

# **Convers3**

**Documentation version 3.08**

**Par Eric DAVID : [vtopo@free.fr](mailto:vtopo@free.fr)**

<b>1</b>	<b>INSTALLATION DE CONVERS</b> .....	<b>2</b>
<b>2</b>	<b>UTILISATION DE CONVERS</b> .....	<b>2</b>
2.1	FENETRE PRINCIPALE .....	2
2.1.1	<i>Convention de saisie et d'affichage des coordonnées</i> .....	2
2.1.2	<i>Saisie des coordonnées à convertir</i> .....	2
2.1.3	<i>Affichage des coordonnées converties</i> .....	2
2.1.4	<i>Conversions par grille</i> .....	3
2.2	PARAMETRAGE .....	3
2.2.1	<i>Ellipsoïdes</i> .....	3
2.2.2	<i>Systèmes géodésiques</i> .....	3
2.2.3	<i>Projections</i> .....	4
2.3	TABLEAU DE CONVERSIONS .....	4
2.4	SAUVEGARDE AU FORMAT XML .....	5
<b>3</b>	<b>CONVERSAPI3.DLL</b> .....	<b>6</b>
3.1	ENUMERATIONS .....	6
3.1.1	<i>Unités</i> .....	6
3.1.2	<i>Méridiens</i> .....	6
3.2	CLASSE LOCATION .....	6
3.2.1	<i>Données membres</i> .....	6
3.2.2	<i>Méthodes</i> .....	6
3.2.3	<i>Exemple</i> .....	7
3.3	STRUCTURE PROJECTIONTYPE .....	7
3.3.1	<i>Données Membres</i> .....	7
3.3.2	<i>Exemple</i> .....	8
3.4	CLASSE CONVERSION .....	8
3.4.1	<i>Données membres</i> .....	8
3.4.2	<i>Méthodes</i> .....	8
3.5	UTILISATION DE CONVERSAPI3.DLL EN C# .....	11
3.6	UTILISATION DE CONVERSAPI3.DLL AVEC EXCEL .....	12
3.7	TYPES DE PROJECTIONS RECONNUES PAR CONVERSAPI3.DLL .....	13

# 1 Installation de Convers

Exécutez Convers3.msi.

Convers nécessite le framework .NET 3.5, il est donc possible que celui-ci soit téléchargé pendant l'installation de Convers.

## 2 Utilisation de Convers

Convers est un logiciel de conversion de coordonnées géographiques qui permet de convertir des coordonnées exprimées en Latitude/Longitude dans différents systèmes géodésiques ou bien en X/Y dans différentes projections.

### 2.1 Fenêtre principale

La fenêtre principale de Convers permet d'effectuer les conversions proprement dites. Elle est séparée en 2 zones distinctes : Départ et Arrivée. La zone Départ permet de saisir les coordonnées à convertir et certains paramètres, la zone Arrivée est utilisée pour afficher les coordonnées converties. Au milieu se trouve une zone permettant de sélectionner une éventuelle grille de conversion. Au lancement de Convers, les options validées sont celles correspondant à la dernière conversion effectuée.

#### 2.1.1 Convention de saisie et d'affichage des coordonnées

Les latitudes Nord seront positives et les latitudes Sud négatives (de  $-90^\circ$  à  $90^\circ$  ou de  $-100\text{gr}$  à  $100\text{gr}$ ).

Les longitudes Est seront positives et les longitudes Ouest négatives (de  $-180^\circ$  à  $180^\circ$  ou de  $-200\text{gr}$  à  $200\text{gr}$ ).

Les zones UTM Nord seront positives et les zones UTM Sud négatives (de  $-60$  à  $60$ ).

#### 2.1.2 Saisie des coordonnées à convertir

Sélectionner le système géodésique ou la projection correspondant aux coordonnées à convertir dans la liste intitulée Départ. Convers affiche alors automatiquement le type des données à saisir : Latitude/Longitude ou X/Y en fonction de la sélection.

Sélectionner le système géodésique ou la projection correspondant aux coordonnées converties dans la liste intitulée Arrivée.

Si les données sont de type Latitude/Longitude, choisir l'unité des angles et le méridien d'origine (Greenwich ou Paris) dans le cadre intitulé Paramètres avant de commencer la saisie. Dans le cas où l'on a saisi des valeurs avant de choisir l'unité des angles, Convers n'effectue aucune conversion d'angles.

Si les données sont de type X/Y, choisir l'unité des distances dans le dans le cadre intitulé Paramètres avant de commencer la saisie. Dans le cas où l'on a saisi des valeurs avant de choisir l'unité des distances, Convers n'effectue aucune conversion de distance.

Plusieurs unités sont à votre disposition pour saisir les latitudes et longitudes. Le tableau suivant récapitule ces possibilités. L'exemple considère le même angle de  $44^\circ 35' 17.634''$  converti en différentes unités.

Unité	Format	Exemple	Saisie
Degrés.MnSec	Degrés Minutes Secondes	$44^\circ 35' 17.634''$	44.3517634
Degrés.Mn	Degrés Minutes	$44^\circ 35.294'$	44.35294
Degrés déc.	Degrés décimaux	$44^\circ 5882317$	44.5882317
Grades	Grades	49.5424796gr	49.5424796
Radians	Radians	0.7782115	0.7782115

On peut alors saisir les coordonnées à convertir, que Convers convertit automatiquement au fur et à mesure de la saisie, si les données saisies sont correctes. En cas d'erreur de saisie, des indications apparaissent à droite de la valeur erronée.

#### 2.1.3 Affichage des coordonnées converties

Les coordonnées s'affichent en Latitude/Longitude ou en X/Y selon la sélection effectuée dans la liste Arrivée.

Il est possible de sélectionner l'unité d'affichage des angles ou des distances et le méridien d'origine dans le cadre intitulé Unités (la conversion se fait automatiquement) et le méridien d'origine (Greenwich ou Paris) dans le cadre intitulé Origine.

L'affichage du résultat est formaté selon les conventions en vigueur.

## 2.1.4 Conversions par grille

Pour les conversions entre certains systèmes géodésiques, il existe des grilles prédéfinies, permettant un calcul plus précis. Convers3 permet l'utilisation de ces grilles. Pour effectuer une conversion par grille, il suffit de sélectionner la grille souhaitée dans la liste déroulante, le nom de la grille se composant du nom abrégé des systèmes géodésiques concernés séparés par `_`. Par exemple pour passer de NTF à RGF93 ou de Lambert II étendu à Lambert 93, on peut utiliser la grille NTF\_RGF93. La conversion est possible dans les deux sens. Si la grille correspond aux données à convertir, et que le point concerné se trouve dans la zone délimitée par la grille alors cette dernière est utilisée et l'indication  s'affiche à droite de la grille. Dans le cas contraire, la conversion est réalisée par la méthode standard et aucune indication de s'affiche.

Il est possible d'ajouter de nouvelles grilles en fonctions de vos besoins, Convers3 acceptant les grilles au format NTV2 (extension `.gsb`). L'ajout se fait simplement en renommant le fichier `.gsb` selon la convention précisée plus haut et en le copiant dans le répertoire de Convers3, `C:\Program Files\Convers3`. Il faut ensuite relancer Convers3.

## 2.2 Paramétrage

Convers est livré avec la plupart des ellipsoïdes couramment utilisées, les systèmes géodésiques : WGS84, ED50, NTF, RGF93 et les projections : Lambert (zones I, II, II étendu, III, IV), Lambert Carto (zones I, II, III), Lambert 93 (y compris les zones CC42 à CC50), UTM / WGS84, UTM / ED50. D'autres ellipsoïdes, systèmes géodésiques et projections peuvent être ajoutés pour correspondre à vos besoins. Les données sont stockées dans une base de données Access qui se trouve dans le même dossier que Convers : `Convers.mdb`

### 2.2.1 Ellipsoïdes

Le bouton Ellipsoïdes affiche une boîte de dialogue permettant l'ajout, la modification et la suppression d'ellipsoïdes. Cette boîte de dialogue contient les paramètres nécessaires à la définition d'un ellipsoïde.

<b>Nom abrégé</b>	Nom abrégé de l'ellipsoïde
<b>Nom complet</b>	Nom complet de l'ellipsoïde
<b>Demi grand axe (A)</b>	Demi grand axe de l'ellipsoïde en mètres
<b>Demi petit axe (B)</b>	Demi petit axe de l'ellipsoïde en mètres
<b>Aplatissement inverse (1/F)</b>	Aplatissement inverse de l'ellipsoïde

2 boutons à droite des zones de saisie de B et 1/F permettent de choisir la valeur à saisir, A, B, et 1/F étant liés par une relation mathématique. Selon les documentations, on trouve les valeurs de A et B ou de A et 1/F.

Le bouton Nouveau crée un nouvel ellipsoïde. Il faut saisir un nom abrégé qui doit être unique.

Le bouton Supprimer supprime l'ellipsoïde sélectionné.

Le bouton Enregistrer enregistre les modifications.

Le bouton Annuler annule les modifications.

### 2.2.2 Systèmes géodésiques

Le bouton Datums affiche une boîte de dialogue permettant l'ajout, la modification et la suppression de systèmes géodésiques. Cette boîte de dialogue contient les paramètres nécessaires à la définition d'un système géodésique.

<b>Nom abrégé</b>	Nom abrégé du système.
<b>Nom complet</b>	Nom complet du système. Il apparaît dans les listes Départ et Arrivée.
<b>Ellipsoïde de référence</b>	Nom de l'ellipsoïde de référence.
<b>Tx</b>	Translation en X pour passer au système pivot (WGS84). En mètres
<b>Ty</b>	Translation en Y pour passer au système pivot (WGS84). En mètres
<b>Tz</b>	Translation en Z pour passer au système pivot (WGS84). En mètres
<b>Rx</b>	Rotation en X pour passer au système pivot (WGS84). En secondes d'angle
<b>Ry</b>	Rotation en Y pour passer au système pivot (WGS84). En secondes d'angle
<b>Rz</b>	Rotation en Z pour passer au système pivot (WGS84). En secondes d'angle
<b>Echelle</b>	Facteur d'échelle pour passer au système pivot (WGS84). En parties par million (ppm)

Pour utiliser les transformations à 7 paramètres, il faut cocher la case 'Conversion à 7 paramètres'.

Les rotations entre les systèmes doivent être saisis selon la convention IERS (les axes du système d'arrivée sont ramenés parallèles aux axes du système de départ). La plupart des logiciels SIG et des récepteurs GPS utilisent une convention différente (les signes de Rx, Ry et Rz sont alors inversés)

Le bouton Nouveau crée un nouveau système géodésique. Il faut saisir un nom abrégé qui doit être unique.

Le bouton Supprimer supprime le système géodésique sélectionné.  
 Le bouton Enregistrer enregistre les modifications.  
 Le bouton Annuler annule les modifications.

### 2.2.3 Projections

Le bouton Projections affiche une boîte de dialogue permettant l'ajout, la modification et la suppression de projections. Cette boîte de dialogue contient les paramètres nécessaires à la définition d'une projection.

<b>Nom abrégé</b>	Nom abrégé de la projection
<b>Nom complet</b>	Nom complet de la projection. Il apparaît dans les listes Départ et Arrivée.
<b>Système géodésique</b>	Nom abrégé du système géodésique associé à la projection.
<b>Type de projection</b>	Définit la projection utilisée.
<b>Paramètres</b>	Paramètres de la projection. Ils sont variables selon le type de projection. Les noms des paramètres nécessaires apparaissent automatiquement dans le tableau en fonction du type de projection.

Certains paramètres des projections sont des angles. Converse adopte les conventions de saisie suivantes pour la notation des angles dans cette boîte de dialogue :

Angle en degrés minutes secondes : 48°35'54.682" saisir 48.3554682°  
 Angle en degrés minutes : 48°35.91100' saisir 48.3591100m  
 Angle en degrés décimaux : 48.5985228 saisir 48.5985228d  
 Angle en grades : 53.9983586 saisir 53.9983586g  
 Angle en radians : 0.8482042 saisir 0.8482042r

Les longitudes seront toujours exprimées par rapport au méridien de Greenwich.

Le bouton Nouveau crée une nouvelle projection. Il faut saisir un nom abrégé qui doit être unique.  
 Le bouton Supprimer supprime la projection sélectionnée.  
 Le bouton Enregistrer enregistre les modifications.  
 Le bouton Annuler annule les modifications.

### 2.3 Tableau de conversions

Ce tableau permet d'effectuer des conversions en nombre. Le principe de saisie est le même que pour la fenêtre principale, l'affichage peut être soit formaté soit brut, selon que la case « Affichage formaté » soit cochée ou non. Le résultat s'affiche automatiquement lorsque les valeurs nécessaires ont été saisies. Le résultat sera « Err » en cas d'erreur de saisie ou de valeurs erronées. Il est possible de copier (Ctrl + C) ou de coller (Ctrl + V) des données à partir d'un autre logiciel. En cas d'utilisation d'une grille de conversion, la marque  apparaîtra pour chaque ligne ayant utilisé la grille.

Le bouton Ouvrir ouvre un fichier csv dont le format est le suivant :

1<sup>ère</sup> ligne : Depart;code;unité;[méridien]  
 2<sup>ème</sup> ligne : Arrivee;code;unité;[méridien]  
 3<sup>ème</sup> ligne (facultative) : Grille ;grille de conversion  
 3<sup>ème</sup> ou 4<sup>ème</sup> ligne et suivantes : données

Depart et Arrivee (sans accents) sont des mots réservés et doivent figurer en tête des 1<sup>ère</sup> et 2<sup>ème</sup> lignes.

Code est le nom abrégé de la projection ou du système géodésique de départ ou d'arrivée.

Unités : 0 = Degrés Minutes Secondes  
 1 = Degrés Minutes  
 2 = Degrés décimaux  
 3 = Grades  
 4 = Radians  
 5 = Mètres  
 6 = Kilomètres

Méridiens (uniquement si le code correspond à un système géodésique) :  
 1 = Greenwich  
 2 = Paris

Grille : nom de la grille de conversion à utiliser, par exemple NTF\_RGF93.

Les données sont dans l'ordre suivant :

Si le code correspond à un système géodésique : Latitude ;Longitude

Si le code correspond à une projection sans zone : X ;Y

Si le code correspond à une projection avec zone : X ;Y ;Zone

Exemple de fichier csv à convertir :

Depart;UTM.WGS84;5

Arrivee;WGS84;0;1

500000;5500;31

500100;550100;31

Le bouton Enregistrer enregistre un fichier csv. Le format est le même que celui saisi précédemment avec en plus le résultat des calculs qui peuvent être formatés ou bruts.

Exemple de fichier csv converti :

Depart;UTM.WGS84;5

Arrivee;WGS84;0;1

500000;5500000;31;N 49°39'09.2";E 003°00'00.0"

500100;5500100;31;N 49°39'12.4";E 003°00'05.0"

Le bouton Effacer efface la totalité du tableau.

## **2.4 Sauvegarde au format XML**

Cette fonctionnalité permet de sauvegarder la totalité des paramètres de Convers, ellipsoïdes, systèmes géodésiques et projections dans un fichier xml. Elle sert essentiellement aux utilisateurs de ConversApi3.dll (voir plus bas) qui ainsi pourront définir la configuration nécessaire à leurs applications avec Convers et utiliser ConversApi3.dll de manière totalement autonome grâce à sa fonction d'initialisation à partir d'un fichier xml.

## 3 ConversApi3.dll

Convers est livré avec ConversApi3.dll afin de permettre l'utilisation de son moteur de calcul pour effectuer des conversions dans vos propres applications en .Net. Un exemple est fourni en C# dans C:\Program Files\Convers\Exemples\C#. Pour les applications en 64 bits, une version x64 de ConversApi3.dll est disponible dans C:\Program Files\Convers\x64\ConversApi64.zip.

ConversApi3.dll contient un ensemble de classes décrites ci-dessous. Le namespace se nomme ConversApi.

### 3.1 Enumérations

Ces énumérations définissent les unités utilisées pour exprimer les coordonnées et les méridiens de référence.

#### 3.1.1 Unités

```
enum Units { DMS, DMN, DDEC, GRAD, RAD, MT, KM };
```

DMS : Degrés Minutes Secondes.  
DMN : Degrés Minutes.  
DDEC : Degrés décimaux.  
GRAD : Grades.  
RAD : Radians.  
MT : Mètres.  
KM : Kilomètres.

#### 3.1.2 Méridiens

```
enum Meridians { NONE, GREENWICH, PARIS };
```

NONE : Aucun méridien. Utilisé pour des coordonnées de type X / Y.  
GREENWICH : Méridien de Greenwich. Utilisé pour des coordonnées de type Latitude / Longitude.  
PARIS : Méridien de Paris. Utilisé pour des coordonnées de type Latitude / Longitude.

### 3.2 Classe Location

Cette classe correspond à la représentation d'un point avec son système de coordonnées et ses unités. Elle permet le formatage des coordonnées selon les conventions en vigueur.

#### 3.2.1 Données membres

double XLon : Longitude si le point est exprimé en Latitude / Longitude. X si le point est exprimé en X / Y.  
double YLat : Latitude si le point est exprimé en Latitude / Longitude. Y si le point est exprimé en X / Y.  
int Zone : Zone si le point représente des coordonnées UTM.  
char Designator : Désignateur si le point représente des coordonnées UTM.  
String Key : Nom abrégé du système géodésique ou de la projection correspondant au point.  
Units Unit : Unité dans laquelle est exprimé le point  
Meridians Mer : Méridien de référence du point s'il est exprimé en Latitude / Longitude.  
String sXLon : Longitude ou X formaté avec la méthode Format()  
String sYLat : Latitude ou Y formaté avec la méthode Format()  
String sZone : Zone et désignateur formatés avec la méthode Format()

#### 3.2.2 Méthodes

Location()

Constructeur par défaut.

Location(double xLon, double yLat, String key, Units unit, Meridians mer)

Constructeur utilisé pour un point exprimé en Latitude / Longitude.

`double` xLon : Longitude.  
`double` yLat : Latitude.  
`String` key : Nom abrégé du système géodésique.  
`Units` unit : Unité.  
`Meridians` mer : Méridien de référence.

`Location(double xLon, double yLat, String key, Units unit)`

Constructeur utilisé pour un point exprimé en X/Y sans zone .

`double` xLon : X.  
`double` yLat : Y.  
`String` key : Nom abrégé de la projection.  
`Units` unit : Unité.

`Location(double xLon, double yLat, int zone, char designator, String key, Units unit)`

Constructeur utilisé pour un point exprimé en X/Y avec zone (UTM)

`double` xLon : X.  
`double` yLat : Y.  
`int` zone : Zone UTM.  
`char` designator : Désignateur UTM.  
`String` key : Nom abrégé de la projection.  
`Units` unit : Unité.

Si on ne connaît pas le désignateur, il est possible de renseigner uniquement la zone, positive pour l'hémisphère Nord, négative pour l'hémisphère Sud, et de positionner le désignateur à `'\0'` .

`Location(Location from)`

Constructeur à partir d'un autre objet Location.

`Location` from : Objet d'origine

`void` `Format(Conversion conv)`

Formate l'affichage des coordonnées selon les normes en vigueur pour chaque type de données.

`Conversion` conv : Objet Conversion de référence (voir la description de la classe Conversion).

Le résultat se trouve dans les variables `sXLon`, `sXLat`, `sZone`.

### 3.2.3 Exemple

```
Location loc1 = new Location(5.47307, 45.38582, "WGS84", Units.DMS, Meridians.GREENWICH);
Location loc2 = new Location(869.269, 2078.565, "LT2E", Units.KM);
Location loc3 = new Location(717540, 5058899, 31, '\0', "UTMWGS84", Units.MT);
```

```
loc1.Format(conv);
Console.WriteLine(loc1.sYLat + " " + loc1.sXLon); // N 45°34'54.7" E 005°23'43.3"
```

Les systèmes géodésiques et projections `"WGS84"`, `"LT2E"`, `"UTMWGS84"` doivent avoir été définis dans l'objet `conv` (voir la description de la classe Conversion).

## 3.3 Structure ProjectionType

Cette structure donne des informations sur les paramètres des types de projections.

### 3.3.1 Données Membres

`String` Key : Nom abrégé de la projection.

`String` Name : Nom complet de la projection.  
`String[]` Param : Nom des paramètres de la projection.

La classe `Conversion` contient une liste de `ProjectionType` permettant d'obtenir les informations sur toutes les projections reconnues par `ConversApi3.dll`.

### 3.3.2 Exemple

```
foreach (ProjectionType pt in Conv.ProjectionTypes)
{
    Console.WriteLine(pt.Key + " " + pt.Name);

    foreach (String str in pt.Param)
        Console.WriteLine(str);
}
```

## 3.4 Classe Conversion

Cette classe permet d'effectuer les conversions de coordonnées. Elle doit être initialisée soit à partir d'un fichier xml soit à l'aide de méthodes permettant de définir tous les paramètres nécessaires. Elle possède également plusieurs méthodes utilitaires. Les méthodes génèrent des exceptions avec une description explicite en cas d'erreur.

Avant de pouvoir effectuer des conversions, il est nécessaire de paramétrer l'objet `Conversion`, soit avec la méthode `ReadXml`, soit avec les méthodes d'ajout d'ellipsoïdes (`AddEllipsoid`), de systèmes géodésiques (`AddDatum`), et de projections (`AddProjection`).

Les ellipsoïdes devront être ajoutés avant les systèmes géodésiques et les systèmes géodésiques avant les projections, ceci afin de garantir la cohérence des données.

### 3.4.1 Données membres

`List<ProjectionType>` `ProjectionTypes` : Liste des projections reconnues par `ConversApi3.dll`.

### 3.4.2 Méthodes

`Conversion()`

Constructeur.

`void` `AddEllipsoid(String key, String descr)`

Ajoute un ellipsoïde.

`String` key : Nom abrégé de l'ellipsoïde.  
`String` descr : Paramètres sous la forme [nom];[a];[b];[1/f]

nom : Nom complet de l'ellipsoïde.  
a : Demi grand axe en mètres.  
b : Demi petit axe en mètres.  
1/f : Aplatissement inverse.

Si b est différent de 0 alors 1/f doit être égal à 0 et réciproquement, ces 2 valeurs étant dépendantes l'une de l'autre.

`void` `AddEllipsoid(String key, String name, double a, double b, double invF)`

Ajoute un ellipsoïde.

`String` key : Nom abrégé de l'ellipsoïde.  
`String` name : Nom complet de l'ellipsoïde.  
`double` a : Demi grand axe en mètres.  
`double` b : Demi petit axe en mètres.  
`double` invF : Aplatissement inverse.

Si b est différent de 0 alors 1/f doit être égal à 0 et réciproquement, ces 2 valeurs étant dépendantes l'une de l'autre.

```
void AddDatum(String key, String descr)
```

Ajoute un système géodésique.

```
String key      : Nom abrégé du système géodésique.  
String descr   : Paramètres sous la forme [nom];[ellipsoïde];[tx];[ty];[tz];[rx];[ry];[rz];[echelle]
```

```
nom      : Nom complet du système géodésique.  
ellipsoïde : Nom abrégé de l'ellipsoïde de référence.  
tx, ty, tz : Translations en X, Y, Z en mètres.  
rx, ry, rz : Rotations en X, Y, Z en mètres. Uniquement pour les transformations à 7 paramètres.  
echelle  : Facteur d'échelle en ppm (parties par million). Uniquement pour les transformations à 7 paramètres.
```

Pour les transformations à 3 paramètres, il suffit de renseigner nom, ellipsoïde, tx, ty, tz.

```
void AddDatum(String key, String name, String ellips, double tx, double ty, double tz)
```

Ajoute un système géodésique pour des transformations à 3 paramètres.

```
String key      : Nom abrégé du système géodésique.  
String name    : Nom complet du système géodésique.  
String ellips  : Nom abrégé de l'ellipsoïde de référence.  
double tx, ty, tz : Translations en X, Y, Z en mètres.
```

```
void AddDatum(String key, String name, String ellips, double tx, double ty, double tz,  
              double rx, double ry, double rz, double scale)
```

Ajoute un système géodésique pour des transformations à 7 paramètres.

```
String key      : Nom abrégé du système géodésique.  
String name    : Nom complet du système géodésique.  
String ellips  : Nom abrégé de l'ellipsoïde de référence.  
double tx, ty, tz : Translations en X, Y, Z en mètres.  
double rx, ry, rz : Rotations en X, Y, Z en secondes d'angle.  
double scale    : Facteur d'échelle en ppm (parties par million).
```

```
void AddProjection(String key, String descr)
```

Ajoute une projection.

```
String key      : Nom abrégé de la projection.  
String descr   : Paramètres sous la forme [nom];[type];[sysgeo];[p1][p2];... ;[pn]
```

```
nom      : Nom complet de la projection.  
type    : Type de projection. (Voir en 3.5 les projections supportées).  
sysgeo  : Système géodésique de référence.  
p1      : Paramètre 1 de la projection.  
p2      : Paramètre 2 de la projection.  
pn      : Paramètre n de la projection.
```

Les paramètres doivent être dans l'ordre spécifié pour chaque projection (Voir en 3.5). Les valeurs des paramètres doivent respecter les conventions de saisie détaillées en 2.2.3.

```
void AddProjection(String key, String name, String typeProj, String datum, String []param)
```

Ajoute une projection.

```
String key      : Nom abrégé de la projection.  
String name    : Nom complet de la projection.  
String typeProj : Type de projection. (Voir en 3.5 les projections supportées).  
String datum   : Système géodésique de référence.  
String []param : Tableau de paramètres.
```

Les paramètres doivent être dans l'ordre spécifié pour chaque projection (Voir en 3.5). Les valeurs des paramètres doivent respecter les conventions de saisie détaillées en 2.2.3.

`void LoadGrid(String file)`

Charge une grille de conversion.

`String file` : Chemin complet de la grille à charger.

`void Clear()`

Réinitialise l'objet Conversion. Efface les ellipsoïdes, systèmes géodésiques, projections et système géodésique de référence.

`bool Convert(Location from, Location to, String grid = "")`

Effectue une conversion.

`Location from` : Position à convertir.

`Location to` : Position convertie.

`String grid` : Nom de la grille de conversion

Voir la description de la classe Location en 3.2.

La variable `from` doit contenir les coordonnées du point à convertir (XLon et YLat), le nom abrégé du système géodésique ou de la projection dans laquelle sont exprimées les coordonnées (Key), l'unité (Unit), le méridien de référence (Mer) s'il s'agit de coordonnées en Latitude / Longitude, la zone (Zone) s'il s'agit de coordonnées UTM.

La variable `to` doit contenir le nom abrégé du système géodésique ou de la projection dans laquelle seront exprimées les coordonnées (Key), l'unité (Unit), le méridien de référence (Mer) s'il s'agit de coordonnées en Latitude / Longitude. Au retour, les coordonnées (XLon et YLat) seront renseignées, ainsi que la zone et le désignateur s'il s'agit de coordonnées UTM.

La variable `grid` est le nom d'une grille de conversion au format NTV2, qui sera utilisée si les coordonnées sont à l'intérieur de la grille (fichier .gsb). Si cette variable n'est pas renseignée, la conversion sera réalisée sans grille de manière classique.

Valeur de retour : `true` si la grille précisée a été utilisée, `false` sinon ou si aucune grille n'est précisée.

`bool IsDatum(String key)`

Détermine si le nom abrégé `key` correspond à un système géodésique déjà défini dans l'objet Conversion.

`String key` : Nom abrégé.

Valeur de retour : `true` si le nom abrégé correspond à un système géodésique, `false` sinon.

`bool IsEllipsoid(String key)`

Détermine si le nom abrégé `key` correspond à un ellipsoïde déjà défini dans l'objet Conversion.

`String key` : Nom abrégé.

Valeur de retour : `true` si le nom abrégé correspond à un ellipsoïde, `false` sinon.

`bool IsProjection(String key)`

Détermine si le nom abrégé `key` correspond à une projection déjà définie dans l'objet Conversion.

`String key` : Nom abrégé.

Valeur de retour : `true` si le nom abrégé correspond à une projection, `false` sinon.

`bool IsGrid(String grid)`

Détermine si la grille de nom `grid` est chargée en mémoire.

`String grid` : Nom de la grille de conversion.

Valeur de retour : `true` si la grille est chargée, `false` sinon.

```
bool MinMaxZone(String key, out int minZone, out int maxZone)
```

Obtient les zones minimale et maximale d'une projection à zone automatique.

```
String key : Nom abrégé de la projection.  
int minZone : Zone minimale.  
int maxZone : Zone maximale.
```

Si la projection ne correspond pas à une projection avec calcul de zone automatique, `minZone` et `maxZone` vaudront 0.

Valeur de retour : `true` si le nom abrégé correspond à une projection avec calcul de zone automatique, `false` sinon.

```
Dictionary<String,String> ReadXml(String file, bool append = false)
```

Initialise l'objet Conversion à partir d'un fichier xml.

```
String file : Chemin et nom du fichier xml.  
bool append : Mode de lecture (true pour ajouter, false pour initialiser)
```

Le format du fichier est le même que celui obtenu en utilisant la sauvegarde au format xml de Convers (Voir 2.4). En mode initialisation, tous les éléments de l'objet Conversion seront effacés avant l'ajout des nouveaux. En mode ajout les éléments ayant un code déjà existant dans l'objet Conversion ne seront pas ajoutés.

Valeur de retour : Dictionnaire des éléments ajoutés. Key contient le type (E pour les ellipsoïdes, D pour les système géodésiques, P pour les projections) suivi d'un ; et du code de l'élément. Value contient la description des éléments (voir les méthodes AddEllipsoïde, AddDatum, et AddProjections pour la description de chaque élément).

```
void WriteXml(String file)
```

Enregistre les paramètres de l'objet Conversion dans un fichier xml.

```
String file : Chemin et nom du fichier xml.
```

```
bool ZoneAuto(String key)
```

Détermine si la projection de nom abrégé `key` permet le calcul de zone automatique.

```
String key : Nom abrégé de la projection.
```

Valeur de retour : `true` si le nom abrégé correspond à une projection avec calcul de zone automatique, `false` sinon.

### 3.5 Utilisation de ConversApi3.dll en C#

Cet exemple est fourni avec Convers3, il s'ouvre avec Visual Studio (2008 et suivants) : Menu Démarrer de Windows, Convers 3, Exemples, C#, Exemple.sln.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.IO;  
using ConversApi;  
  
namespace Exemple  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Conversion Conv = new Conversion();  
  
            if (File.Exists("Convers.xml")) // Configuration à partir d'un fichier xml  
            {  
                Conv.ReadXml("Convers.xml");  
            }  
            else // Configuration manuelle
```

```

{
    // Ajout des ellipsoïdes
    Conv.AddEllipsoid("WGS84", "WGS 1984", 6378137, 6356752.314245, 0);
    Conv.AddEllipsoid("CLRK80IGN", "Clarke 1880 IGN", 6378249, 6356515, 0);

    // Ajout des systèmes géodésiques
    Conv.AddDatum("WGS84", "World Geodetic System 1984", "WGS84", 0, 0, 0);
    Conv.AddDatum("NTF", "Nouvelle Triangulation Française", "CLRK80IGN", -168, -60, 320);

    // Ajout des projections
    Conv.AddProjection("UTM.WGS84", "Universal Transverse Mercator / WGS84;UTM;WGS84;0.9996");
    Conv.AddProjection("LT2E", "Lambert France zone II étendue;LAMBCC2;NTF;45.5356108°;47.4145652°;
        2.2014025°;52g;6e5;2.2e6");
}

// Point à convertir en Lambert II étendu (kilomètres)
Location from = new Location(869.269, 2078.565, "LT2E", Units.KM);

// Résultat en Lat / Lon WGS84 (degrés minutes secondes, Greenwich)
Location to = new Location(0,0,"WGS84",Units.DMS,Meridians.GREENWICH);

// Conversion et affichage du résultat
Conv.Convert(from, ref to);
from.Format(Conv);
to.Format(Conv);

Console.WriteLine("Lambert II étendu : X = {0} Y = {1}", from.sXLon, from.sYLat);
Console.WriteLine("WGS84                : {0} {1}", to.sYLat, to.sXLon);

// Conversion en UTM
to.Key = "UTM.WGS84";
to.Unit = Units.MT;
to.Mer = Meridians.NONE;

Conv.Convert(from, ref to);
to.Format(Conv);

Console.WriteLine("UTM / WGS84          : {0:00}{1} {2} {3}", to.Zone, to.Designator, to.sXLon, to.sYLat);
Console.Read();
}
}

```

### 3.6 Utilisation de ConversApi3.dll avec Excel

Il est possible d'utiliser ConversApi3.dll en VBA avec Excel. Avant toute utilisation des classes de ConversApi3.dll il peut être nécessaire de la référencer dans Excel bien que normalement le référencement se fasse de manière automatique lors de l'installation de Convers3. Pour cela, ouvrir le module Visual Basic d'Excel puis procéder au référencement grâce au menu 'Outil – Références' afin de référencer C:\Program Files\Convers3\ConversApi3.tlb. Une fois le référencement effectué, l'accès aux fonctionnalités de ConversApi3.dll sera possible.

L'exemple suivant est fourni avec Convers3. Pour l'ouvrir : Menu Démarrer de Windows, Convers 3, Exemples, Excel, Exemple.xls. A partir de la version 2007 d'Excel il faut activer les macros après l'ouverture du fichier. Selon l'endroit où Convers3 sera installé, il faudra peut-être modifier le chemin du fichier Convers.xml passé en paramètre à la fonction ReadXml. En cas d'erreur d'initialisation la fonction renverra 99 pour les coordonnées converties. Un exemple de conversion avec grille NTV2 est également fourni.

```

' Variables globales
Dim Init As Integer
Dim conv As New ConversApi3.Conversion

Public Function ConversCoords3(ByVal dl As Range, ByVal lpszFrom As String, ByVal iUnit1 As Integer, ByVal iMer1 As Integer, ByVal lpszTo As String, ByVal iUnit2 As Integer, ByVal iMer2 As Integer) As Variant

' Déclarations
Dim loc1 As New ConversApi3.Location
Dim loc2 As New ConversApi3.Location

' Initialisation à partir de Convers.xml, à ne faire qu'une seule fois
If Init = 0 Then

```

```

Call conv.ReadXml("C:\Program Files\Convers3\Exemples\Excel\Convers.xml", 0) ' Windows XP

If Not conv.IsDatum("WGS84") Then
    Call conv.ReadXml("C:\Program Files (x86)\Convers3\Exemples\Excel\Convers.xml", 0) ' Windows 7
End If

If conv.IsDatum("WGS84") Then Init = 1 Else Init = -1
End If

If Init = 1 Then

    ' Initialisation du point à convertir

    loc1.xlOn = dl(1, 1)
    loc1.YLat = dl(1, 2)
    If dl.Count > 2 Then loc1.Zone = dl(1, 3)
    loc1.Key = lpszFrom
    loc1.Unit = iUnit1
    loc1.Mer = iMer1

    ' Initialisation du résultat

    loc2.Key = lpszTo
    loc2.Unit = iUnit2
    loc2.Mer = iMer2

    ' Conversion

    Call conv.Convert(loc1, loc2)

    ' Retour du résultat sous forme de matrice

    ConversCoords3 = Array(loc2.xlOn, loc2.YLat, loc2.Zone)
Else
    ConversCoords3 = Array(99, 99, 99)
End If

End Function

```

### 3.7 Types de projections reconnues par *ConversApi3.dll*

#### Cylindrique Equidistante (Equirectangulaire)

Code type : EQUIRECT  
 Paramètres : Méridien central, Latitude d'origine, X à l'origine (m), Y à l'origine (m).

#### Laborde Madagascar

Code type : LABORDE  
 Paramètres : Facteur d'échelle, Azimut, Méridien central, Latitude d'origine, X à l'origine (m), Y à l'origine (m).

#### Lambert Conforme Conique (2 parallèles)

Code type : LAMBCC2  
 Paramètres : 1er parallèle sécant, 2<sup>nd</sup> parallèle sécant, Latitude d'origine, Méridien central, X à l'origine (m), Y à l'origine (m).

#### Lambert Conforme Conique (1 parallèle)

Code type : LAMBCC1  
 Paramètres : Facteur d'échelle à l'origine, Méridien central, Latitude d'origine, X à l'origine (m), Y à l'origine (m)

#### Lambert Conforme Conique (2 parallèles) Belgique 1972

Code type : LAMB72  
 Paramètres : 1er parallèle sécant, 2<sup>nd</sup> parallèle sécant, Latitude d'origine, Méridien central, X à l'origine (m), Y à l'origine (m).

#### Lambert Conique Tronquée

Code type : LAMBCT  
 Paramètres : Facteur d'échelle à l'origine, Méridien central, Latitude d'origine, X à l'origine (m), Y à l'origine (m).

#### Mercator Oblique (Hotine) type A

Code type : MEROBLIQA

Paramètres : Facteur d'échelle, Latitude 1er point, Longitude 1er point, Latitude 2ème point, Longitude 2ème point, Latitude d'origine, X à l'origine (m); Y à l'origine (m)

### **Mercator Oblique (Hotine) type B**

Code type : MEROBLIQB

Paramètres : Facteur d'échelle, Azimut, Méridien central, Latitude d'origine, X à l'origine (m), Y à l'origine (m)

### **Mercator Transverse**

Code type : MERTRANS

Paramètres : Facteur d'échelle, Méridien central, Latitude d'origine, X à l'origine (m), Y à l'origine (m)

### **Miller Cylindrique**

Code type : MILLER

Paramètres : Méridien central, X à l'origine (m), Y à l'origine (m).

### **Stéréographique Equatoriale**

Code type : STEREQUA

Paramètres : Facteur d'échelle, Méridien central, X à l'origine (m), Y à l'origine (m)

### **Stéréographique Oblique**

Code type : STEROBLIQ

Paramètres : Facteur d'échelle, Méridien central, Latitude d'origine, X à l'origine (m), Y à l'origine (m)

### **Suisse (Oblique Conforme Cylindrique)**

Code type : SWISS

Paramètres : Méridien central, Latitude d'origine, X à l'origine (m), Y à l'origine (m)

### **Universal Transverse Mercator (UTM)**

Code type : UTM

Paramètres : Facteur d'échelle.

### **Universal Transverse Mercator (UTM) Zone fixe**

Code type : UTMZONE

Paramètres : Facteur d'échelle, Zone (Nord > 0, Sud < 0).