



Report

TRANSINT .NET / COM library

**12-15
A**

October 2012

Jérôme Ray

Developer's manual



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Bundesamt für Landestopografie swisstopo
Office fédéral de topographie swisstopo
Ufficio federale di topografia swisstopo
Uffizi federal da topografia swisstopo

www.swisstopo.ch

SQS-Zertifikat ISO 9001:2000



TRANSINT .NET / COM library

12-15 A

Jérôme Ray

Developer's manual

October 2012

Table of contents

1	Introduction	1
2	Technical aspects / calculation settings.....	1
3	Minimal system requirements	1
4	Installation	2
5	Using the TRANSINT DLL	3
5.1	Namespace.....	3
5.2	Data types and enumerations.....	3
5.3	Interface	3
5.4	Main class	3
5.5	Methods, returns and exceptions	3
6	Code samples	5
6.1	C# .NET	5
6.2	C++	7
6.3	VB/VBA	8

© 2012 **swisstopo**
Bundesamt für Landestopografie
Office fédéral de topographie
Ufficio federale di topografia
Uffizi federal da topografia
Federal Office of Topography

Redaktion: A. Wiget
Seftigenstrasse 264
CH-3084 Wabern
Tel: +41 31 963 24 69
Fax: +41 31 963 24 59
E-mail: adrian.wiget@swisstopo.ch

swisstopo Report ist die Nachfolgeserie der Reihe "Technische Berichte" in welcher über die wichtigen Arbeiten aus den Bereichen von swisstopo berichtet wird.

swisstopo Report est la suite de la série "Rapports techniques", qui décrivent les projets et activités les plus importants de swisstopo.

swisstopo Report is the follow-up of the series "Technical Report" in which important projects and activities of swisstopo are described.

- Report (A):** **Öffentliche Berichte** (public domain) von swisstopo: Keine Einschränkungen. Weitergabe und Reproduktion mit Quellenangabe erwünscht.
Rapports officiels du swisstopo (domaine public): Sans restrictions. Diffusion et reproduction souhaitée.
Public domain of swisstopo: no restrictions. Duplication and distribution with reference to source.
- Report (B):** **Berichte eigener Arbeiten von swisstopo:** Alle Rechte vorbehalten. Das Copyright bleibt bei der swisstopo. Reproduktion nur mit schriftlicher Bewilligung der swisstopo.
Rapports sur les travaux internes du swisstopo: Les droits de reproduction (copyright) restent au swisstopo. Reproduction seulement avec l'autorisation écrite du swisstopo.
Reports on internal activities: All rights reserved. The copyright remains with swisstopo. Reproduction requires the written permission by swisstopo.
- Report (C):** **Berichte von Arbeiten für Dritte** (Werkvertrag): Alle Rechte vorbehalten. Das Copyright geht an den Auftraggeber über. Jede Weitergabe, vollständige oder teilweise Reproduktion oder Speicherung in elektronischen Medien nur mit schriftlichem Einverständnis des Auftraggebers möglich.
Rapports sur les travaux mandatés par des tiers: Les droits de reproduction appartiennent au mandataire. Toute diffusion ou reproduction (même partielle possible uniquement avec l'autorisation écrite du mandataire).
Reports commissioned by third parties: All rights reserved. The copyright remains with the client. Duplication, reproduction and storage on electronic media of any or all parts of the report require written permission by the client.
- Report (D):** **Interne Berichte der Bereiche.** Alle Rechte vorbehalten. Nur für internen Gebrauch.
Rapport interne des domaines. Les droits de reproduction (copyright) restent au swisstopo. Exclusivement à usage interne.
Internal reports. All rights reserved. Only for internal use.

1 Introduction

“TRANSINT” is a calculation software for planimetric transformation and interpolation of points, allowing the comparison of geodetic networks or adjustments with existing fixed points. This software offers several methods of transformation and also allows the robust adjustment. The coordinates transformation with different models can be performed with manually defined or automatically calculated parameters.

TRANSINT can be used to compute local adjustments (interpolations) for the Swiss reference frame change LV03↔LV95.

Since 2012, TRANSINT is available as a calculation module for the new geodetic software “GeoSuite”:

→ <http://www.swisstopo.ch/geosuite>

And for technical aspects about Swiss reference frames and the new frame LV95:

→ <http://www.swisstopo.ch/lv95>

The TRANSINT DLL allows software developers to implement TRANSINT computing in their own applications through .NET or COM components (.NET or C++ applications, VBA macros, further DLL or plug-ins, etc.).

At this time, the only calculation available is the **interpolation based on a regular grid or an assembly of grids**. The transformation based on a triangular network (“FINELTRA” algorithm) is intended to be provided in this DLL later. Further methods will perhaps be added subsequently, depending on customer requests or mandates.

2 Technical aspects / calculation settings

The TRANSINT DLL can only compute **two-dimensional** (planimetric) data.

The interpolation data or calculation parameters must be provided in an external file:

- **Interpolation grid**

A regular grid defined with its coordinate extents, its size (number of columns and rows), its resolution (mesh size) and the x (east) and y (north) components or corrections.

This grid has to be provided in the **GeoSuite Grid format** (*.gsg), which can be generated with the TRANSINT calculation module of the desktop software GeoSuite.

If you have several grid files to use, there are two ways to use them:

1. Make a GeoSuite assembly file (*.gsg format) with TRANSINT calculation module. Ask the provider of the grids for it.
2. Manage the two grids separately: you'll have to instantiate two interpolation objects (see also “CreateGridInterpolation” method) and decide yourself which grid has to be used (or which object to call) for which coordinates: For example, if a grid has been provided for a specific commune, canton or region, the choice of the grid or the check if the grid is applicable or not has to be performed at your side (in the client application).

3 Minimal system requirements

- Hardware: PC Pentium 400 MHz or compatible with 96 Mb RAM (or more, depending on operating system)
- Operating system: Microsoft Windows XP, Vista, 7 or Server 20XX

- Software: **Microsoft .NET Framework**¹ 3.5 (or further)

4 Installation

swisstopo delivers a standard **Windows Installer setup** (MSI²) which installs all required files and automatically registers the DLL in the system. You just have to execute it and follow the on screen indications.

To optimize file size and download time, we didn't integrate neither the Microsoft Windows Installer engine nor the Microsoft Windows .NET Framework. If they aren't already installed on your computer, they will be automatically downloaded³ and installed.

After the installation's process, you'll find a shortcut to a PDF version of this manual in the Windows start menu.

If you want to create your own distribution or install the DLL on another computer without our setup, you'll have to copy the files as follows:

1. Copy the program library file into the chosen installation directory:

- **swisstopoTransintLib.dll**

Note: you have to register the DLL file properly with the "regasm" tool (see below), and eventually update the reference in your source code (e.g. "#import" instruction in C++).

2. If the TRANSINT DLL is called through COM components (and not directly referenced in a .NET application), you have to register the .NET assembly into the Windows registry by running the following command:

regasm swisstopoTransintLib.dll /codebase /tlb

Warning: on systems where Microsoft .NET Framework SDK (or Visual Studio) isn't installed (but only the .NET Framework redistributable), you'll have to type the full "regasm" path, e.g. "C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\regasm.exe ...". You will also need to specify explicitly "...Framework64\..." for the 64 bits version.

Note: this registration can be done automatically with most of the setup software, as they support natively the "COM Interop" registration.

3. Copy the additional "DotNumerics" library, required for some interpolation algorithms, into the chosen installation directory:

- **DotNumerics.dll**

4. Copy the additional transformation parameters or binary datasets (e.g. **interpolation grid or triangular network**) to the right target folder, depending on your source code (see chapter 5).

Microsoft .NET Framework 3.5 has to be installed on the target computer. It is automatically provided on all Microsoft Windows operating system since Vista. A free version can be downloaded and redistributed for Windows XP.

If you develop in **C++** with Microsoft Visual Studio, you also have to distribute the "**Microsoft Visual C++ Redistributable Package**" corresponding to the Visual Studio version used to compile the final binary.

¹ Microsoft .NET Framework is not included in the provided setup to limit the file size, but it can be downloaded for free on the Microsoft website (<http://www.microsoft.com/downloads/>)

² "Microsoft System Installer" or "Windows Installer"

³ They also can be obtained through the Windows Update service or downloaded manually : <http://www.microsoft.com/downloads/>

5 Using the TRANSINT DLL

This DLL has been developed with Visual Studio 2010 and is entirely written in **C# managed code**. It is compatible with **x86 and x64** architectures (compiled with “Any CPU” setting). This library offers COM interoperability for use with unmanaged code.

5.1 Namespace

In the .NET environment, the namespace is “swisstopo.geodesy.dll”. If you are working with C++, you have to use “swisstopoTransintLib” to access the DLL.

Examples are provided in the chapter 6.

5.2 Data types and enumerations

All the computations are made with “double” (signed IEEE 64-bit double-precision floating point numbers) values to ensure the best accuracy.

Only standard data types are used in the TRANSINT DLL and it is fully compatible with C++ environments.

To improve the clearness of the code and reduce the mistake sources, a **user-defined enumeration** has been created to identify the grid interpolation function (method “CreateGridInterpolation” below):

```
public enum GridInterpolationFunction
{
    Automatic = 0,
    Bilinear = 1,
    Biquadratic = 2,
    Bicubic = 3
}
```

5.3 Interface

The main (and only) interface is named “ITransint” and contains 2 methods.

```
[Guid("6E8AC103-BFD4-4DE0-B7EB-24CE6A247FBA")]
public interface ITransint
{
    void CreateGridInterpolation(string gridFilePath,
                                GridInterpolationFunction interpolationFunction);
    void InterpolatePoint(ref double east, ref double north);
}
```

5.4 Main class

The TRANSINT DLL provides only the main class, named “Transint”, which implements the “ITransint” interface described above.

The constructor is standard and doesn’t take any argument:

```
public Transint()
```

5.5 Methods, returns and exceptions

The “Transint” class contains two public methods:

- **CreateGridInterpolation**
Create a new interpolation object, which will provide interpolation calculation based on a regular grid.
- **InterpolatePoint:**
Transformation of two-dimensional coordinates with the interpolation method previously defined.

CreateGridInterpolation

This method needs 2 arguments:

```
void CreateGridInterpolation(string gridFilePath,  
                             GridInterpolationFunction interpolationFunction)
```

1. **gridFilePath:**
Full path of the regular interpolation grid to use, in GeoSuite Grid binary format (*.gsg).
2. **interpolationFunction:**
The interpolation function to use to compute the x and y values at the specified coordinates. It has to be a value of the "GridInterpolationFunction" enum (see 5.2 above) or his corresponding integer value.

***Warning:** the interpolation function can have been defined in the grid file. In this case, this parameter **is locked and cannot be overridden** and an exception will be thrown. If the interpolation function is predefined, the enum value "**GridInterpolationFunction.Automatic**" has to be used.*

***Note:** if the interpolation function is set to "**Automatic**" and it has not been defined in the GeoSuite Grid file, "**Bilinear**" is used.*

This method does not returns anything (void).

Following **exceptions** can be thrown:

- **FileNotFoundException**
(HRESULT COR_E_FILENOTFOUND = 0x80070002 = -2147024894)
The specified interpolation grid file doesn't exists or cannot be found (e.g. removable media cannot be accessed). Refer to exception description for more details.
- **FormatException**
(HRESULT FORMAT = 0x80131537 = -2146233033)
The specified interpolation grid file cannot be loaded: invalid file or unsupported grid file format. This exception could also be raised if the file cannot be accessed (file locked or security restriction). Refer to exception description for more details.
- **AccessViolationException**
(HRESULT COR_E_NULLREFERENCE = 0x80004003 = -2147467261)
The specified interpolation function cannot be used because this parameter has been defined in the grid file and is locked. In this case, "GridInterpolationFunction.Automatic" must be used. Refer to exception description for more details.

InterpolatePoint

This method needs 2 arguments:

```
void InterpolatePoint(ref double east, ref double north)
```

1. **east:**
Easting coordinate in meters [m]. This argument is passed by reference: it means that **the input value is replaced by the output coordinate** after the processing.
2. **north:**
Northing coordinate in meters [m]. This argument is passed by reference: it means that **the input value is replaced by the output coordinate** after the processing.

This method does not return anything (void).

Following **exceptions** can be thrown:

- **MethodAccessException**
(HRESULT COR_E_METHODACCESS = 0x80131510 = -2146233072)

This method cannot be called if no interpolation has been previously created: "CreateGridInterpolation" has to be called once before "InterpolatePoint". Refer to exception description for more details.
- **FormatException**
(HRESULT FORMAT = 0x80131537 = -2146233033)

The specified interpolation grid file (method "CreateGridInterpolation") could not be loaded: invalid file or unsupported grid file format. This exception could also be raised if the file cannot be accessed (file locked or security restriction). Refer to exception description for more details.
- **ArgumentOutOfRangeException**
(HRESULT COR_E_ARGUMENTOUTOFRANGE = 0x80131502 = -2146233086)

The specified coordinates are outside of the specified interpolation grid: the interpolation is undefined for this location. Refer to exception description for more details.

6 Code samples

6.1 C# .NET

Before you can use the DLL, you have to add a .NET reference to "**swisstopo TRANSINT library 1.0 (09.2012)**" (**swisstopoTransintLib.dll**). The namespace is "**swisstopo.geodesy.dll**".

Example of C# code:

```
// Reference to the TRANSINT DLL
using swisstopo.geodesy.dll;
...

// Instantiate a TRANSINT object
Transint oTransint = new Transint();

// Define the interpolation to use
// (regular grid with predefined interpolation method)
try
{
    oTransint.CreateGridInterpolation(@"C:\MyApp\Data\MyGrid.gsg",
        GridInterpolationFunction.Automatic);
}
catch (FileNotFoundException)
{
    // Exception if grid file could not be found
}
catch (FormatException)
{
    // Exception if grid file could not be loaded
}
```

```
catch (AccessViolationException)
{
    // Exception if the interpolation function is readonly
    // (it has been defined in the grid file and is now locked)
}
catch (Exception)
{
    // Other exception (should not happen)
}

// Input coordinates: read in a file, got from a textbox,
// or obtained through another method or library...
double east = 601000.0; // Input (local) east
double north = 197500.0; // Input (local) north

// Transform coordinates to output system (global)
try
{
    oTransint.InterpolatePoint(ref east, ref north);
}
catch (MethodAccessException)
{
    // Exception if CreateGridInterpolation hasn't been called yet
    // or if didn't work properly
}
catch (FormatException)
{
    // Exception if grid file could not be loaded
}
catch (ArgumentOutOfRangeException)
{
    // Exception if the point is outside of the interpolation grid
    // A possibility could be to apply a mean translation (shift)
}
catch (Exception)
{
    // Other exception (should not happen)
}

// Make something with the result: the output coordinate values
// are now stored in "east" and "north" variables
...

```

6.2 C++

Before you can use the DLL, you have to import the COM type library (*.tlb) “swisstopoTransintLib.tlb” and initialize the COM library with “CoInitialize”.

Example of C++ code:

```
// Reference to TRANSINT DLL
#import <swisstopoTransintLib.tlb>4
...

// Initialize COM connection
CoInitialize(NULL);

// Create a pointer to the Transint class from DLL
swisstopoTransintLib::ITransintPtr
    pTransint(__uuidof(swisstopoTransintLib::Transint));

// Define the interpolation to use
// (regular grid with predefined interpolation method)
try
{
    pTransint->CreateGridInterpolation(
        "C:\\MyApp\\Data\\MyGrid.gsg",
        swisstopoTransintLib::GridInterpolationFunction_Automatic);
}
catch (_com_error& comErr)
{
    switch (comErr.Error())
    {
    case -2147024894:
        // Exception if grid file could not be found
        break;
    case -2146233033:
        // Exception if grid file could not be loaded
        break;
    case -2147467261:
        // Exception if the interpolation function is readonly
        // (has been defined in the grid file and is now locked)
        break;
    default:
        // Other exception (should not happen)
        break;
    }
}
```

⁴ If the TRANSINT DLL and the type library are correctly registered in the system with “regasm” (see chapter 4). If not or if you are experiencing problems when you try to instantiate a Transint object, you can import the TLB file giving the full path (e.g. “C:\MyApp\swisstopoTransintLib.tlb”). This will be more “secure”, but less flexible.

```

// Input coordinates: read in a file, got from a textbox,
// or obtained through another method or library...
double east = 601000.0; // Input (local) east
double north = 197500.0; // Input (local) north

// Transform coordinates to output system (global)
try
{
    pTransint->InterpolatePoint(&east, &north);
}
catch (_com_error& comErr)
{
    switch (comErr.Error())
    {
        case -2146233072:
            // Exception if CreateGridInterpolation hasn't been called
            // yet or if it didn't work properly

            break;
        case -2146233033:
            // Exception if grid file could not be loaded
            break;
        case -2146233086:
            // Exception if the point is outside of the grid
            // A possibility could be to apply a translation (shift)
            break;
        default:
            // Other exception (should not happen)
            break;
    }

    // Make something with the result: the output coordinate values
    // are now stored in "east" and "north" variables
    ...

```

6.3 VB/VBA

Before you can use the component, you have to add a reference to **“swisstopo TRANSINT library 1.0 (09.2012)” (swisstopoTransintLib.tlb)**.

Example of VB code:

```

'Create a new Transint object
Dim oTransintLib As New swisstopoTransintLib.Transint

'Define the interpolation to use
'(regular grid with predefined interpolation method)
On Error Resume Next

oTransintLib.CreateGridInterpolation "C:\MyApp\Data\MyGrid.gsg",
GridInterpolationFunction_Automatic

```

```
Select Case Err.Number
  Case 0
    'OK
  Case -2147024894
    'Exception if grid file could not be found
  Case -2146233033
    'Exception if grid file could not be loaded
  Case -2147467261
    'Exception if the interpolation function is readonly
    '(has been defined in the grid file and is now locked)
  Case Else
    'Other exception (should not happen)
End Select

On Error GoTo 0

'Input coordinates: read in a file, got from a textbox,
'or obtained through another method or library...
Dim east As Double, north As Double
east = 601000# 'Input (local) east
north = 197500# 'Input (local) north

'Transform coordinates to output system (global)
On Error Resume Next
oTransintLib.InterpolatePoint east, north

Select Case Err.Number
  Case 0
    'OK
  Case -2146233072
    ' Exception if CreateGridInterpolation hasn't been called
    // yet or if it didn't work properly
  Case -2146233033
    'Exception if grid file could not be loaded
  Case -2147467261
    'Exception if the interpolation function is readonly
    '(has been defined in the grid file and is now locked)
  Case Else
    'Other exception (should not happen)
End Select

'Make something with the result: the output coordinate values
'are now stored in "east" and "north" variables
...
```

Wabern, October 2012

Federal Office of Topography swisstopo
Geodetic Developments and Contractual Work
Jérôme Ray