

ChronoMap Developer Kit
TourSolver Developer Kit



**REDUCE YOUR TIME-TO-MARKET
GEOAPPLICATIONS**

OUR KNOW-HOW AT YOUR SERVICE

WHO ARE THE PROGRAMMING COMPONENTS DESIGNED FOR?

SOFTWARE PUBLISHERS AND SERVICE PROVIDERS

- > You developed an enterprise sales management solution and you want to visualize clients on a map, make simple itinerary calculation or propose catchments area analysis tool → ChronoMap Developer Kit.
- > You developed an order entry software and you want to add a delivery scheduling module → TourSolver Developer Kit.
- > You sell a vehicle tracking service and your customers are asking for a complementary routes optimization service → TourSolver Developer Kit.

INTEGRATORS AND LARGE ACCOUNTS

- > You want to automate your whole logistics supply chain by integrating the delivery scheduling into the enterprise resource and manufacturing planning software → TourSolver Developer Kit.
- > You set up an Intranet service to display incident locations, to find the closest resource that can help and calculate the access time and the best route to go there → ChronoMap Developer Kit.

COMPETITIVE ADVANTAGES

ALL-IN-ONE

- > All modules you need: geocoding, itineraries, mileage charts, accessibility, optimization, display.
- > OPTI-TIME provides in standard the road database modeled for OPTI-TIME components format: the DataPack.

EASY TO INTEGRATE, EASY TO IMPLEMENT

- > Our know-how and expertise made accessible through simple methods.
- > Faster time to market via our numerous samples for major development environments.
- > Easy to deploy.
- > Same tools from the desktop computer to multiprocessor servers.
- > Affordable: prices scalable with the architecture.
- > International availability: using standard databases sold by Navteq and Tele Atlas. Also available for other databases using OPTI-TIME formatting tools.

TourSolver Developer Kit

ChronoMap Developer Kit

Optimization	Display	Routing	Geocoding
Network Modeler			
 Road network databases		 Magellan Exclusivity	 Traffic statistics

Our programming components are multi-platform development tools that allow you to easily incorporate mapping and routing function into your applications. Don't reinvent the wheel—use our know-how to quickly build route optimization, scheduling, store locators, or driving-direction solutions.

TWO PROGRAMMING COMPONENTS

CHRONOMAP DEVELOPER KIT:

the mapping and routing functions

- > geocoding
- > itineraries, mileage charts, routing, itinerary optimization
- > accessibility, restricting, assignment
- > vehicle profile management
- > display (Windows only)

TOURSOLVER DEVELOPER KIT:

includes ChronoMap Developer Kit and adds scheduling and route optimization functions

- > customer constraints management
- > resource parameters management
- > scheduling and route optimization
- > reporting and results exports

ChronoMap Developer Kit and TourSolver Developer Kit are available both on Windows and Linux environment *.

* The display module is not provided for Linux. The developer can use anyone of the numerous free software of the market like MapServer.

MAP DATABASES

DATAPACK: MARKET LEADING DATA, READY FOR USE

Our tools work with standard market leading databases Navteq & TeleAtlas.

We format these databases into a "DataPack" that contains all necessary files for the routing, geocoding and display modules.

Other databases may be used by converting them into a compatible format with our Network Modeler application.

Geographical areas (up to an entire continent) can be employed while maintaining geocoding and routing accuracy at the street number level.



NAVTEQ



DISPLAY

(Windows environment only)

- > Included in our DataPack.
- > Display locations, itineraries, routes, driving time areas, sectors.
- > Management of the area to be displayed, of the zoom, of the centering.
- > Possibility of using other map display tools. Some functions of the routing module allow to get the coordinates of the locations, the routes or the areas to be displayed.

For Linux the developer can use anyone of the numerous free software of the market like MapServer.

```
...
m_Geocoder.GetWayPoint (CountryId, ref x, ref y);
m_Mapper.CenterX = x;
m_Mapper.CenterY = y;
m_Mapper.Zoom = ZoomLevelConstants.cmCity;
m_Mapper.MapWidth = m_Map.Width;
m_Mapper.MapHeight = m_Map.Height;
m_Mapper.ExportMap + "cmdk_pic.png", ImageTypeConstants.cmPNG);
if (m_Map.Image != null) m_Map.Image.Dispose();
m_Map.Image = Image.FromFile ("cmdk_pic.png");
m_Map.Refresh();
...
```

NETWORK MODELER: USING OTHER DATABASES

- > Network Modeler allows you to convert map databases (other than Navteq and Tele Atlas) into the format required by our components.
- > Network Modeler provides only the files required by the routing and geocoding modules.
- > Available in command line.



GEOCODING

Geocoding

- > number level: longitude, latitude starting from an address
- > inverse: address starting from longitude, latitude coordinates
- > point of interest: shopping center, airport, gas station, industrial park, etc.

Information provided

- > most relevant address
- > list of suggestions in decreasing order by relevance

Accuracy and score

- > geocoding accuracy: number, nearest number, street, block, postal code, city
- > indicates confidence level of the results

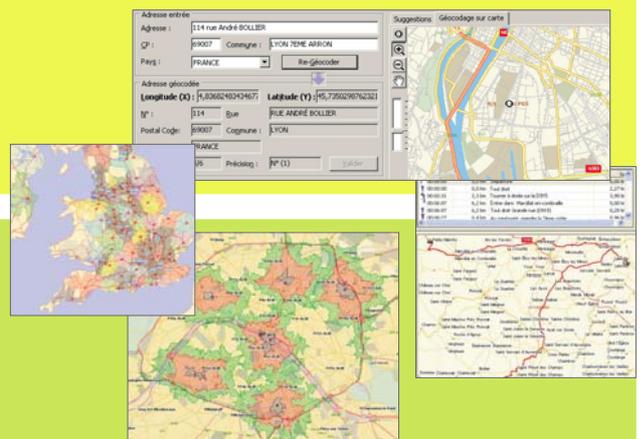
* subject to the availability of appropriate databases for those countries.

Analysis and correction

- > address field parsing
- > addresses correction: abbreviations, troublesome characters, spelling mistakes

International and multilingual

- > UNICODE
- > international address management
- > multilingual address management
- > non-latin character management *



ROUTING AND ACCESSIBILITY

Itinerary calculations

- > start, arrival and stop points
- > shortest or quickest route
- > stop point order optimization

Mileage charts

- > distance matrices in seconds and meters while taking into account the travel direction
- > partial distance matrices: nearest points and/or points located within a given distance or access time

Roadbooks

- > detailed roadbooks giving directions, maneuvers and traffic signs

Driving access time (isochrones)

- > areas covered starting from/or going to a given point
- > available in drive-time or in distance
- > single or multiple points
- > analyze single or multiple areas simultaneously

Redistricting

- > redistricting of a geographical area based on a set of points
- > going to/or starting from the points
- > redistricting based on drive-time or distance

Assignment

- > assignment of locations to the nearest point from a set of points
- > calculation in time or distance
- > distance and access time to the nearest point

Vehicle profile management

- > vehicles characteristics and behavior: speeds on different types of roads, authorized or restricted roads, rather taking major roads, avoiding toll ways, ...
- > list of streets or areas that should be avoided
- > "Truck attributes" (bridges height, maximum weight, hazardous materials, etc.)*
- > historical traffic speeds *

* depends on availability of this content for the database used.

MINIMAL CODE

OPTIMIZATION (TourSolver Developer Kit only)

The optimization module adds the simple concepts of "Orders" and "Resources" for calculating the lowest global cost of servicing a group of customers with your available resources.

Optimization is based on:

- > Customer and resource locations accurately determined by the geocoding module.
- > Drive-times and distances as calculated by the routing module.
- > Customers constraints: delivery time windows, quantities, delivery days, etc.
- > Resources parameters: work time, costs, skills, etc.

"Orders" object constraints

- > Characteristics: identifier, particularities, street side *etc.
- > Quantities: up to 8 quantity dimensions in parallel.
- > Time management: delivery time windows, possible days, duration and frequency.
- > Link to resources: assigning or excluding one or several resources.
- > Existing routes evaluation.

"Resources" object parameters

- > Characteristics: skills, attributes.
- > Capacities: up to 8 capacity dimensions, loading time, multi-warehouses, etc.
- > Time management: work hours and days, overtime hours, lunch break, briefing/debriefing.
- > Costs management: hourly cost, distance cost,

fixed costs, etc.

- > Overnight routes.
- > Driving time regulations.
- > Numerous combinations: departure/arrival at the depot, from home, nights out on the road.

Controls and warnings

- > Data format and consistency checks.
- > Format, data inconsistency or inaccessible customers warnings.

Exporting, drawing and printing

- > Export routes in xml-xls, txt and html formats.
- > Detailed roadbooks, driving directions.
- > Gantt chart for one or all routes.
- > Specific objects for directly drawing and printing routes in mapping environments.



```
using TourSolverDeveloperKit;
public partial class TeaserForm : Form {
    private TourSolver m_TourSolver = new TourSolver();
    private Locations m_XYs = null;
    private Directions m_Dirs = null;
    private void m_GoGeocoder_Click(object sender, EventArgs e) {
        StreetAddresses magellanHQs = new StreetAddresses();
        magellanHQs.Add("hq-Grenoble", "710", "Rue Aristides Berges", "38330", "", "Montbonnot", "", "", "", "FRANCE");
        magellanHQs.Add("hq-Paris", "88", "Rue Robespierre", "93100", "", "Montreuil", "", "", "", "FRANCE");
        m_XYs = m_TourSolver.Geocoder.StreetsToLocations(magellanHQs);
    }
    private void m_GoRouter_Click(object sender, EventArgs e) {
        m_TourSolver.Router.VehicleProfile.VehicleType = VehicleTypeConstants.cmCar;
        m_TourSolver.Router.VehicleProfile.PreferMajorRoads = true;
        m_TourSolver.Router.VehicleProfile.AvoidTollWays = true;
        Routes m2mRoutes = m_TourSolver.Router.GetRoutes(m_XYs, 0, false);
        m_Dirs = m2mRoutes.get_Directions(1);
        m_Report.VirtualListSize = m_curDirs.Count;
        DriveZones ThirtyMinSalesZone = m_TourSolver.Router.GetDriveZones(magellanHQs, 1800, false, true);
    }
    private void m_GoSolver_Click(object sender, EventArgs e) {
        Solver m_Solver = m_TourSolver.Solver;
        m_Solver.Orders.Add("tsOrderQuantity=10", "Water/Grenoble", m_XYs.Get_X(0), m_XYs.Get_Y(0), "");
        m_Solver.Orders.Add("tsOrderQuantity=20", "Water/Paris", m_XYs.Get_X(1), m_XYs.Get_Y(1), "");
        m_Solver.Resources.Add("tsResourceCapacity=30", "Water Provider", 3.54, 45.45, "");
        Locations problemXYs = m_Solver.GetDataLocations();
```

```
m_Solver.UseDriveTimeMatrix(m_TourSolver.Router.GetDriveTimeMatrix(problemXYs, problemXYs, -1));
if (m_Solver.CheckConfiguration() != CheckStatusConstants.tsOK) {
    m_Solver.Solve(-1);
    m_Solver.Orders.get_Details("Water/Grenoble", StopDetailConstants.tsStopStartTime);
    m_Solver.Orders.get_Details("Water/Paris", StopDetailConstants.tsStopStartTime);
    m_Dirs = m_Solver.Resources.get_Routes("Water Provider", 0).get_Directions(1);
    m_Report.VirtualListSize = m_curDirs.Count;
}
else MessageBox.Show(m_Solver.Resources.get_Warnings("Water Provider").get_Message(0))
}
private void m_map_MouseMove(object sender, MouseEventArgs e) {
    Double longitude, latitude;
    m_TourSolver.Mapper.PixelsToWayPoint(e.X, e.Y, longitude, latitude);
    m_latitudeStrip.Text = latitude.ToString();
    m_longitudeStrip.Text = longitude.ToString();
    Locations clicked = new Locations();
    clicked.Add(longitude, latitude, false, "");
    m_addressStrip.Text = m_TourSolver.Geocoder.LocationsToStreets(clicked).get_FullStreet(0);
}
private void m_Report_RetrieveVirtualItem(object sender, RetrieveVirtualItemEventArgs e) {
    e.Item.Text = Convert.ToString(m_curDirs.get_ElapsedDistance(e.ItemIndex), new KilometerFormattedDistance());
    e.Item.SubItems.Add(m_curDirs.get_Instruction(e.ItemIndex));
    e.Item.SubItems.Add(m_curDirs.get_Toward(e.ItemIndex));
    e.Item.ImageIndex = m_curDirs.get_DirectionType(e.ItemIndex);
}
```

OPTI-TIME INNOVATION: HISTORICAL TRAFFIC SPEEDS*

By incorporating historical traffic speeds into your applications, your results will be more accurate. Using a database of traffic speeds, we know the average speed of traffic flow on a particular road segment for each hour of the day. Take advantage of our innovation and your results will be a reflection of what actually occurs on the road.



OBJECT MODELS



TECHNICAL CHARACTERISTICS

ChronoMap Developer Kit and TourSolver Developer Kit are available for both Windows and Linux.

WINDOWS

Windows 2000 or Windows XP O/S required.

Available as 32-bit COM objects. Compatible with ASP, Visual Basic or any other environment supporting COM objects like Borland Delphi and .NET.

They are also available as a C DLL and can be used with the provided C++ Framework for a direct access to the objects without going through COM.

LINUX

The Linux components are delivered as C libraries with a C++ interface. They are compiled and tested on Debian Sarge 3.1 for the i386 and x86_64 architectures.

REQUIRED MEMORY

512 MB minimum. The developer can manage the memory used by the components.

SAMPLES

Numerous samples are provided:

- > Geocoding
- > Itinerary & Accessibility
- > Optimization
- > Internet Routing

For the following environments:

- > VisualBasic 6.0
- > C#
- > VB.NET
- > ASP.NET/C#
- > C++/Visual Studio
- > C++/gcc

* depends on countries.