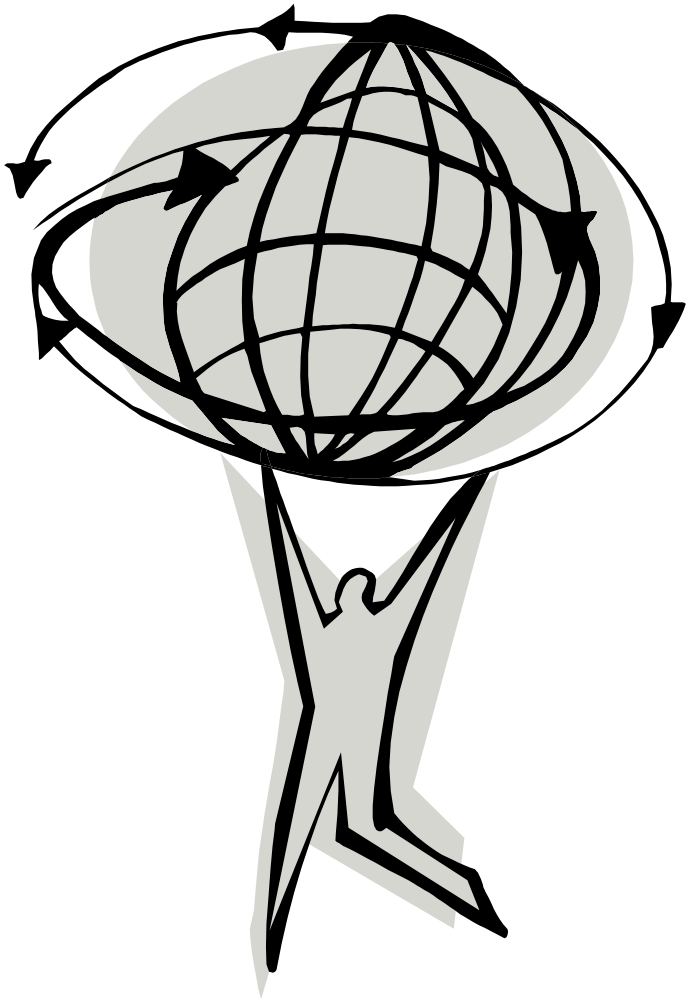
UnderTow Software

MapTivate v3.0 Reference Manual



Undertow Software, Inc. 26011 Lauren Drive Channahon, Illinois 60410
Ph (815) 521-9950 Fx (815) 521-9952 www.undertowsoftware.com

MapTivate OCX Version 3.0
License Agreement

PLEASE READ THE FOLLOWING TERMS AND CONDITIONS. THIS LEGAL DOCUMENT IS AN AGREEMENT BETWEEN YOU AND UNDERTOW SOFTWARE, INC.(THE "COMPANY"). BY USING THIS PRODUCT, YOU ARE AGREEING TO BE BOUND BY THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THESE TERMS AND CONDITIONS, PROMPTLY RETURN THE COMPLETE PACKAGE TO THE PLACE YOU OBTAINED IT FOR A REFUND.

1. GRANT OF LICENSE: In consideration of your payment of this single-user license fee, which is the price you paid for this product, and your agreement to abide by the terms and conditions of this Agreement, the COMPANY grants to you a nonexclusive right to use, develop with, and display a copy of the software developers toolkit MapTivate v3.0 (hereinafter the "SOFTWARE") on a single computer (i.e., with a single CPU) at a single location so long as you comply with the terms of this Agreement. The Company reserves all rights not expressly granted to you under this Agreement.

2. OWNERSHIP OF SOFTWARE: You own only the magnetic or physical media (the enclosed disks) on which the SOFTWARE is recorded or fixed, but the Company retains all the rights, title, and ownership to the SOFTWARE recorded on the original disk, downloaded files, and all subsequent copies of the SOFTWARE, regardless of the form or media on which the original or other copies may exist. The license is not a sale of the SOFTWARE or any copy to you.

3. SOFTWARE COPY RESTRICTIONS: This SOFTWARE and the accompanying printed materials and user manual (the "DOCUMENTATION") are the subject of copyright. You may not copy or distribute the DOCUMENTATION or the SOFTWARE, for any reason other than to make a single copy of the SOFTWARE for backup or archival purposes only. You may be held responsible for any copying or copyright infringement which is directly or indirectly caused or encouraged by your failure to abide by the terms of this restriction.

4. MAP RESTRICTIONS: Images created with this SOFTWARE are the subject of copyright. You may include maps as a supplement to any printed or electronic document(s) from which you do not directly or indirectly receive profit or consideration for. You may not distribute map images with any publication or product that is intended to be an atlas or a collection of maps or is in any other way considered a map or map document. You may not distribute map images generated, in whole or in part, by the SOFTWARE directly or indirectly over the internet or a network. You may be held responsible for any copying or copyright infringement which is directly or indirectly caused or encouraged by your failure to abide by the terms of this restriction.

5. USE RESTRICTIONS: You may not install the SOFTWARE for network use, or otherwise use it on or by more than one computer or computer terminal at the same time, unless you have a specific Network or Multi-user License, in which case you cannot use the SOFTWARE on a network with more users than what you are licensed for. You may physically or electronically transfer the SOFTWARE from one computer to another provided that the SOFTWARE remains on only one computer at a time. You may copy sections of the mapping database onto your hard drive for the purpose of improved program performance. You may not distribute copies of the SOFTWARE or DOCUMENTATION to others. You may not reverse engineer, disassemble, decompile, adapt or modify the SOFTWARE. **AN ADDITIONAL LICENSING AGREEMENT IS REQUIRED BY THE COMPANY TO DISTRIBUTE THE SOFTWARE OR FOR USE NOT EXPRESSLY GRANTED IN THIS DOCUMENT.**

6. TRANSFER RESTRICTIONS: The enclosed SOFTWARE is licensed only to you and may not be transferred to any one else without the prior written consent of the COMPANY. Any unauthorized transfer of the SOFTWARE shall result in the immediate termination of this Agreement.

7. TERMINATION: This license is effective until terminated by the COMPANY. This license will terminate automatically without notice from the COMPANY and become null and void if you fail to comply with any provisions or limitations of this license. Upon termination, you shall destroy the DOCUMENTATION and all copies of the SOFTWARE. All provisions of this Agreement as to warranties, limitation of liability, remedies or damages, and our ownership rights shall survive termination.

8. MISCELLANEOUS: The Agreement shall be construed in accordance with the laws of the United States of America and the State of Illinois and shall benefit the COMPANY, its affiliates, and assignees.

LIMITED WARRANTY

The COMPANY warrants that the disc on which the SOFTWARE is distributed shall, for a period of thirty (30) days from delivery (the "Warranty Period") be free, in normal use, from defects in material and workmanship. The COMPANY will have no responsibility to replace any disc that has been damaged by accident, abuse or misapplication. If, during the Warranty Period, a defect in the disc appears, you may return the disc to the COMPANY for replacement. The COMPANY does not warrant the completeness or accuracy of any information contained in the map database or that the use of the SOFTWARE will meet your needs or that its use will be uninterrupted or error free. The foregoing constitutes your sole and exclusive remedy for breach by the COMPANY of any warranties (express or implied) made under this License.

EXCEPT FOR THE WARRANTIES SET FORTH ABOVE, THE PRODUCT IS LICENSED "AS IS," AND THE COMPANY DISCLAIMS ANY AND ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-FRINGEMENT. NO COMPANY DEALER, DISTRIBUTOR, AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATION OR ADDITION TO THE FOREGOING WARRANTY.

The information contained within the software has been derived from public records and other sources independent of the COMPANY. While the COMPANY has endeavored to include only information that it believes to be reliable and current it makes no representations or warranties as to the accuracy or completeness of such information or as to any changes that may have occurred subsequent to the date that such information was first obtained by the COMPANY. In addition, actual routing and street and driving conditions may change at any time. Therefore, the user is advised not to rely upon this program as your primary source of navigation or while in the process of driving, or operating any motor vehicle. Furthermore, critical decisions should not be exclusively made based upon the SOFTWARE's generated images, data or information.

Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations and exclusions may not apply to you. This warranty gives you specific legal rights, and you may also have other rights, which vary from state to state. The COMPANY may, from time to time, revise or update the SOFTWARE including DOCUMENTATION, and in so doing, incurs no obligation to furnish such revisions or updates to the customer.

Table of Contents

1. Introduction 21

2. General Control Description 21

Properties Page 22

Zooming Techniques 22

Record Structures 23

Enumerations 24

3. MapTivate Object - Properties 24

AlignBar:TxAlignBar 24

Angle:Integer 24

Sample Code (Delphi 5) 24

AutoPaint:Boolean 25

Sample Code (Delphi 5) 25

AutoQuery:Boolean 25

Sample Code (Delphi 5) 25

Background:IBackground 26

BevelInner:TxBevelCut 26

BevelOuter:TxBevelCut 26

BevelWidth:Integer 26

Sample Code (Delphi 5) 26

BorderStyle:TxBorderStyle 26

BorderWidth:Integer 26

BorderColor:Integer 26

Sample Code (Delphi 5) 27

Config:TxConfig 27

Sample Code (Delphi 5) 27

Cursor:Integer 27

Enabled:Boolean 28

Extents:TRExtents 28

Sample Code (Delphi 5) 28

FileName:String 28

Grid:Boolean 28

Sample Code (Delphi 5) 28

Group:Integer 28

LastGeo:string....28
Sample Code (Delphi 5)....28
MapCenterX:Double....29
Sample Code (Delphi 5)....29
MapCenterY:Double....29
Sample Code (Delphi 5)....29
MapBottom:Double....29
Sample Code (Delphi 5)....29
MapLeft:Double....30
Sample Code (Delphi 5)....30
MapRight:Double....30
Sample Code (Delphi 5)....31
MapTop:Double....31
Sample Code (Delphi 5)....31
MapMode:TxMapMode....31
Sample Code (Delphi 5)....32
MapName:String....32
Sample Code (Delphi 5)....32
MaxZoomScale:Double....32
Sample Code (Delphi 5)....32
MinZoomScale:Double....32
Sample Code (Delphi 5)....32
Sample Code (Visual Basic)....33
OverExtend:Integer....33
Sample Code (Delphi 5)....33
PaintOrder:TxPaintOrder....34
Sample Code (Delphi 5)....34
PixelSIze:Double....34
Sample Code (Delphi 5)....34
RestoreView:Boolean....35
Sample Code (Delphi 5)....35
RunLicense:Boolean....35
ScaleDatum:Double....35
Sample Code (Delphi 5)....35
ShowProgress:Boolean....35
ThemeCount:Integer;....35
Sample Code (Delphi 5)....36

Themes[v:Variant]:ITRTheme....36
Sample Code (Delphi 5)....36
Tolerance:Integer....36
ToolBarMode:TxToolBarMode....36
Units:TxUnit....37
Sample Code (Delphi 5)....37
UserCAD:ITRCAD....37
UserDraw:ITRDraw....37
UserPaint:ITRPaint....37
ViewExtents:TRExtents....37
Sample Code (Delphi 5)....37
ViewMrg:ITRViewMgr....38
Visible:Boolean....38
ZoomPixels:Integer....38
Sample Code (Delphi 5)....38
ZoomScale:Double....38
Sample Code (Delphi 5)....38

4. MapTivate Object - Methods....39

AboutBox()....39
Sample Code (Delphi 5)....39
Sample Code (Visual Basic)....40
ClearMap()....40
Sample Code (Delphi 5)....40
Sample Code (Visual Basic)....40
ClearThemes()....40
Sample Code (Delphi 5)....40
Sample Code (Visual Basic)....41
ConnectTheme(S:String; Option:LongInt):TRTheme.....41
Sample Code (Delphi 5)....41
Sample Code (Visual Basic)....42
DeleteTheme(T:TRTheme):LongInt....42
Sample Code (Delphi 5)....42
Sample Code (Visual Basic)....42
ExecRegister(n:bstr):Integer....42
Sample Code (Visual Basic)....43
FindClosestObjectClose();....43
FindFirstClosestObject(Group:Integer; MaxPoints:LongInt; RefPoint:TRPoint;
Distance:Double;CurPoint:TRPoint):TSMObject;....43

Sample Code (Visual Basic) 44
FindNextClosestObject(Distance:Double;CurPoint:TRPoint):TSMObject; 44
FindGeoFirst(S:String):String; 45
Sample Code (Visual Basic) 46
FindGeoNext:String; 46
FindGeoClose:String; 46
FindObject(X,Y:double); 47
Sample Code (Delphi 5) 47
Sample Code (Visual Basic) 47
FindPlaceFirst(S:String):String; 48
Sample Code (Visual Basic) 48
FindPlaceNext:String; 48
FindPlaceClose:String; 48
FindZipFirst(S:String):String; 49
Sample Code (Visual Basic) 49
FindZipNext:String; 49
FindZipClose:String; 49
GeoParse(F:Field; S:String):String; 50
GetArea(P:TXObject):double 50
Sample Code (Delphi 5) 50
GetCentroid(P:TXObject):Trpoint; 51
Sample Code (Delphi 5) 51
GetClosestPoint(P:TXObject; T:TRPoint; Dist:Double):TRPoint 51
Sample Code (Delphi 5) 51
GetCoords(Var x,y:Double) 52
Sample Code (Delphi 5) 52
Sample Code (Visual Basic) 52
GetDistance(X1,Y1,X2,Y2:double):Double; 53
Sample Code (Delphi 5) 53
Sample Code (Visual Basic) 53
GetPerimeter(p:txobject):double; 53
Sample Code (Delphi 5) 53
GetPoint(P:TXObject,Fraction:double):TRPoint; 54
Sample Code (Delphi 5) 54
GotoView(TV:TviewRec) 54
Sample Code (Delphi 5) 54
Sample Code (Visual Basic) 55
HeadsUp(x1,y1,x2,y2:double):LongInt 55

Sample Code (Delphi 5) 55
Sample Code (Visual Basic) 55
LoadProfile(S:String) 56
Sample Code (Delphi 5) 56
Sample Code (Visual Basic) 56
LogOff(FileName:String) 56
Sample Code (Visual Basic) 56
LogOn(FileName:String; Options:LongInt) 56
Sample Code (Visual Basic) 57
MapToScreen(Xi,Yi:double; Xo,Yo:Integer) 57
Sample Code (Delphi 5) 57
NewTheme():TRTheme; 58
Sample Code (Delphi 5) 58
PaintToDC(DC:LongInt; X,Y,W,H:double; Option:Integer); 58
Sample Code (Delphi 5) 59
Sample Code (Visual Basic) 59
RedrawMap(); 59
Sample Code (Delphi 5) 59
RefreshMap(); 59
Sample Code (Delphi 5) 59
SaveToBMP(Filename:String; Format:Integer); 60
Sample Code (Delphi 5) 60
Sample Code (Visual Basic) 60
SavetoGIF(FileName:String); 60
Sample Code (Visual Basic) 60
SaveToJPG(FileName:String; Format, Quality:Integer) 60
Sample Code (Delphi 5) 61
Sample Code (Visual Basic) 61
SaveProfile(S:String) 61
Sample Code (Delphi 5) 61
ScreenToMap(Xi,Yi:Integer; Xo,Yo:double) 61
Sample Code (Delphi 5) 61
SearchDlg() 62
ThemeToBottom(N:Integer):Integer; 62
Sample Code (Delphi 5) 62
ThemeDown(Index:Integer):LongInt; 62
Sample Code (Delphi 5) 62
ThemeToTop(Index:Integer):Integer 62
Sample Code (Delphi 5) 63

ThemeUp(Index:Integer):LongInt; 63
Sample Code (Delphi 5) 63
ThemeManager() 63
XYString(FormatString:WideString):Str 64
Sample Code (Delphi 5) 66
ZoomCenter(X,Y:Double) 66
Sample Code (Delphi 5) 66
Sample Code (Visual Basic) 66
ZoomDouble() 67
Sample Code (Delphi 5) 67
ZoomExtents() 67
Sample Code (Delphi 5) 67
ZoomHalf() 67
Sample Code (Delphi 5) 67
ZoomPan(Dir:TxPan) 67
Sample Code (Delphi 5) 67
ZoomPrevious() 68
Sample Code (Delphi 5) 68
ZoomScrnRect(X1,Y1,X2,Y2:Integer); 68
Sample Code (Delphi 5) 68
ZoomMapRect(X1,Y1,X2,Y2:double) 68
Sample Code (Delphi 5) 68
ZoomTheme(T:ITRTheme) 69
Sample Code (Delphi 5) 69

5. MapTivate Object - Events 69

OnAutoFindObject(T:TRTheme; Index:LongInt; Query:String) 69
Sample Code (Visual Basic) 69
OnClick() 69
Sample Code (Delphi 5) 69
onDblClick() 70
onDistance(Sender: TObject; Current, Total: Double); 70
Sample Code (Delphi 5) 70
OnFindObject (Sender: TObject; var Theme: ITRTheme; index: Integer; const Query: WideString); 70
Sample Code (Delphi 5) 70
OnFindUserObject (Sender: TObject; const value: ITRObject); 70
Sample Code (Delphi 5) 71
onMessage(Sender: TObject; Code: Integer; const Msg: WideString); 71

Sample Code (Visual Basic) 71
onMouseDown (Sender: TObject; Button:TxMouseButton; shift: TxShiftState; x, y: Integer; Xcord, Ycord: Double);....72
OnMouseMove(Sender: TObject; shift: TxShiftState;Nx,Ny:Integer; Rx,Ry:Double) 72
Sample Code (Delphi 5)....72
onMouseUp(Sender: TObject; Button:TxMouseButton; shift: TxShiftState; x, y: Integer; Xcord, Ycord: Double); 72
OnPaintBefore(Sender: TObject; dc, w, h: Integer);....73
OnPaintAfter(Sender: TObject; dc, w, h: Integer); 73
Sample Code (Delphi 5)....73
OnResize() 73
Sample Code (Delphi 5)....73
OnSearch (Sender: TObject; const Theme: ITRTheme; index: Integer; const Data: WideString); 73
Sample Code (Delphi 5)....73
OnThemeList()....74
Sample Code (Delphi 5)....74
6. ITRTheme Object 74
AddStyle():TRStyle; 74
Sample Code (Delphi 5)....74
ClearStyles() 74
Sample Code (Delphi 5)....75
ConvertToBMP(FileName:String)....75
Sample Code (Delphi 5)....75
ConvertToTXF(FileName:String)....75
Sample Code (Delphi 5)....75
Count:Integer 75
Sample Code (Delphi 5)....75
Delete() 76
Sample Code (Delphi 5)....76
DeleteStyle(N:Integer)....76
Sample Code (Delphi 5)....76
Enabled:Boolean....76
EnumerateStyles(Index:Integer; Options:Integer) 76
Sample Code (Delphi 5)....76
Sample Code (Visual Basic) 77
Extents:ITRExtents;....77
Sample Code (Delphi 5)....77

FileName:String 77
Sample Code (Delphi 5) 77
FindClosestObjectClose(); 77
FindFirstClosestObject(MaxPoints:LongInt; RefPoint:TRPoint; Distance:Double;CurPoint:TRPoint):TSMObject; 78
Sample Code (Visual Basic) 78
FindNextClosestObject(Distance:Double;CurPoint:TRPoint):TSMObject; 79
GetObjectArea(N:Integer):Double; 79
GetObjectCentroid(N:Integer):TRPoint; 79
GetObjectClosestPoint(N:LongInt; tp:LongInt; Pt:TRPoint; Dist:Double):TRPoint; 79
GetObjectData(N:Integer):TXObject; 80
Sample Code (Delphi 5) 80
GetObjectPerimeter(N:Integer):Double; 80
GetObjectPoint(N:Integer; Fraction:double):TRPoint; 80
GetSmObject(N:Integer):TSmObject; 80
Sample Code (Visual Basic) 80
InsertStyle(N:LongInt); 82
LoadFromFile(FileName:String) 82
Sample Code (Delphi 5) 82
Lower:Double 83
Sample Code (Delphi 5) 83
Name:String 84
Sample Code (Delphi 5) 84
Priority:Integer 84
Sample Code (Delphi 5) 84
SaveToFile(FileName:String) 84
Sample Code (Delphi 5) 84
SearchData(Ctriterion1, criterion2, Criterion3:String); 84
Sample Code (Visual Basic) 85
StyleCount:Integer 85
Styles[Index:Integer]:ITRStyle 85
Sample Code (Delphi 5) 85
Sample Code (Visual Basic) 85
Upper:Double 86
Sample Code (Delphi 5) 86
Visible:Boolean 86

7. ITRStyle Object 86

Brush:ITRBrush 86
Sample Code (Delphi 5) 86
Clear() 87
Dialog() 87
Enabled:Boolean 87
Font:ITRFont 87
Sample Code (Delphi 5) 87
FormatStr:String 87
Sample Code (Delphi 5) 89
Sample Code (Visual Basic) 89
ID:Integer 90
Sample Code (Delphi 5) 90
Lower:Double 90
Mark:ITRMark 90
Sample Code (Delphi 5) 90
Name:String 90
Sample Code (Delphi 5) 91
Pen:ITRPen 91
Sample Code (Delphi 5) 91
RuleStr:String 91
Sample Code (Delphi 5) 92
QueryStr:String 92
Sample Code (Delphi 5) 94
Upper:Double 94
Visible:Boolean 94

8. ITRPen Object 94

BackColor:Integer 94
Sample Code (Delphi 5) 94
Color:Integer 94
Sample Code (Delphi 5) 95
Sample Code (Visual Basic) 95
Mode:Integer 95
OuterWidth:Integer 95
Sample Code (Delphi 5) 95
Style:Integer; 96
Sample Code (Delphi 5) 96
Width:Integer 96

Sample Code (Delphi 5) 97
Sample Code (Visual Basic) 97

9. ITRBrush Object 97

BackColor:Integer 97
Sample Code (Delphi 5) 97
BitMap:TRBitMap 97
Sample Code (Delphi 5) 97
Color:Integer 98
Sample Code (Delphi 5) 98
Sample Code (Visual Basic) 98
Mode:Integer 99
Style:Integer 99
Sample Code (Delphi 5) 99
Transparent:Boolean 100

10.ITRFont Object 100

Alignment:Integer 100
Sample Code (Delphi 5) 100
Angle:Integer 101
Sample Code (Delphi 5) 101
BackColor:Integer 101
Sample Code (Delphi 5) 101
Casing:Integer 101
Sample Code (Visual Basic) 102
CollisionDetection:Boolean 102
Color:Integer 102
Sample Code (Delphi 5) 102
Frame:Boolean 102
Sample Code (Visual Basic) 102
Height:Integer 103
Sample Code (Delphi 5) 103
Lower:Double 103
Sample Code (Delphi 5) 103
Name:String 103
Sample Code (Delphi 5) 103
Sample Code (Visual Basic) 104
OffsetX:Integer 104
OffsetY:Integer 104

Style:Integer 104
Sample Code (Delphi 5) 105
Upper:double 105
Sample Code (Delphi 5) 105
UseOnce:Boolean 105
Visible:Boolean 106
11.ITRMark Object 106
BorderColor:Integer 106
Sample Code (Delphi 5) 106
BitMap:TRBitMap 107
Sample Code (Delphi 5) 107
Color:Integer 107
Sample Code (Delphi 5) 107
SymbolFont:String 107
Sample Code (Delphi 5) 108
Lower:double 108
Size:Integer 108
Sample Code (Delphi 5) 108
Style:Integer 109
Sample Code (Delphi 5) 109
Symbol:LongInt 109
Sample Code (Delphi 5) 109
Upper:Double 110
Visible:Boolean 110
VisFilter:Integer 110
Sample Code (Delphi 5) 110
12.ITRCAD Object 111
AddStyle():Integer; 111
Clear() 111
Sample Code (Delphi 5) 111
ClearStyles() 111
Sample Code (Delphi 5) 111
Count:Integer 111
Sample Code (Delphi 5) 112
CreateEllipse(X,Y:Double; R:Double; AspRatio:Double); 112
Sample Code (Delphi 5) 112

Sample Code (Visual Basic) 112
CreateLine(X1,Y1,X2,Y2: Double) 112
Sample Code (Delphi 5) 112
Sample Code (Visual Basic) 113
CreatePoint(X,Y: double) 113
Sample Code (Delphi 5) 113
Sample Code (Visual Basic) 113
CreatePolygon(var Points:PtTRPoint; Var Counts:Integer; Num:Integer) 114
Sample Code (Delphi 5) 114
Sample Code (Visual Basic) 114
CreatePolyLine(var Points:PtTRPoint; Var Counts:Integer; Num:Integer) 115
Sample Code (Delphi 5) 115
Sample Code (Visual Basic) 116
CreateRectangle(X1,Y1,X2,Y2: double) 116
Sample Code (Delphi 5) 116
CreateRegPolygon(X1,Y1,X2,Y2:double; N:Integer) 117
Sample Code (Delphi 5) 117
Sample Code (Visual Basic) 117
CreateText(x,y:double; s:widestring) 117
Sample Code (Delphi 5) 117
Sample Code (Visual Basic) 118
DeleteObject(Index, Num:Integer); 118
Sample Code (Delphi 5) 118
DeleteStyle(N:Integer) 118
Sample Code (Delphi 5) 118
Enabled:Boolean 119
Extents:TRExtents 119
Sample Code (Delphi 5) 119
InsertStyle(N:LongInt); 119
Items[Index:Integer]:TRCADObject 119
Sample Code (Delphi 5) 119
LoadFromFile(FileName:Widestring); 120
Sample Code (Delphi 5) 120
Lower:Double 120
SaveToFile 120
Sample Code (Delphi 5) 120
SelectAll(Option:Integer) 120
Sample Code (Delphi 5) 120
SelectDelete() 120

Sample Code (Delphi 5) 121
SelectedStyle:integer 121
SelectPoint(X,Y:Double, Option:Integer) 121
Sample Code (Delphi 5) 121
SelectRange(Index,Num:Integer; Option:Integer); 121
Sample Code (Delphi 5) 121
SelectRectangle(X1,y1,x2,y2: Double; Option:Integer); 122
Sample Code (Delphi 5) 122
SelectToBack() 122
Sample Code (Delphi 5) 122
SelectToFront() 122
Sample Code (Delphi 5) 122
SelectToStyle(N:Integer) 122
Sample Code (Delphi 5) 122
StyleCount:Integer 123
StyleNum:Integer 123
Sample Code (Delphi 5) 123
Styles[Index:Integer]:ITRStyle 123
Sample Code (Delphi 5) 123
Upper:Double 124
Visible:Boolean 124

13.ITRPaint Object (MapTivate.UserPaint) 124

Brush:ITRBrush 124
Sample Code (Delphi 5) 124
SmCircle(X,Y:Double; R:Double; Aspect:Double) 125
Sample Code (Delphi 5) 125
Sample Code (Visual Basic) 125
SmDisk(X,Y:Double, R:Integer) 126
Sample Code (Delphi 5) 126
Sample Code (Visual Basic) 126
Font:ITRFont 126
Sample Code (Delphi 5) 126
SmLine(X1,Y1,X2,Y2:Double) 127
Sample Code (Delphi 5) 127
Sample Code (Visual Basic) 127
Mark:ITRMark 128
Sample Code (Delphi 5) 128
Sample Code (Visual Basic) 128

Pen:ITRPen 128
Sample Code (Delphi 5) 128
SmPoint(X,Y: Double); 129
Sample Code (Delphi 5) 129
SmPolyPoint(Var Points:TRPoint; Var Counts:Integer; N:Integer) 130
Sample Code (Delphi 5) 130
SmPolyLine(Var Points:TRPoint; Var Counts:Integer; N:Integer) 130
Sample Code (Delphi 5) 131
Sample Code (Visual Basic) 131
SmPolygon(Var Points:TRPoint; Var Counts:Integer; N:Integer) 132
Sample Code (Delphi 5) 132
SmRectangle(X1,Y1,X2,Y2:Double) 132
Sample Code (Delphi 5) 133
SmRing(X,Y:Double, R,Delta:Integer) 133
Sample Code (Delphi 5) 133
Sample Code (Visual Basic) 133
SmSphere(X,Y:Double; Radius:Integer) 134
Sample Code (Delphi 5) 134
SmText(X,Y:Double; S:String) 134
Sample Code (Delphi 5) 134

14. TRDrawObject 135

BW:Boolean 135
Caption:String 135
Sample Code (Delphi 5) 135
Enabled:Boolean 136
ID:LongInt 136
Index:Integer 136
Lower:Double 136
Sample Code (Delphi 5) 136
SecCaption:String 136
SecStyle:Integer 137
Selected:Boolean 137
Sample Code (Delphi 5) 137
Style:Integer 137
Sample Code (Delphi 5) 137
Upper:Double 137
Sample Code (Delphi 5) 137

Visible:Boolean....137
X:Double....138
Sample Code (Delphi 5)....138
Y:Double....138
Sample Code (Delphi 5)....138
15. TRDraw Object (MapTivate.UserDraw) 138
AddStyle():Integer; 138
Clear() 139
Sample Code (Delphi 5)....139
ClearStyles() 139
Count:Integer 139
Sample Code (Delphi 5)....139
Delete(Index,Num:Integer) 139
Sample Code (Delphi 5)....139
DeleteStyle(N:Integer) 139
Sample Code (Delphi 5)....140
Enabled:Boolean 140
Extents:TRExtents 140
Sample Code (Delphi 5)....140
FindObjectAtPoint(X,Y:Double); 140
Sample Code (Delphi 5)....140
FindObjectInRect(X1,Y1,X2,Y2:Double); 141
Sample Code (Delphi 5)....141
InsertStyle(N:LongInt); 141
Objects[Index:Integer]:ITRDrawObject 141
Sample Code (Delphi 5)....141
LoadFromFile(FileName:String); 141
Sample Code (Delphi 5)....141
Lower:Double 142
NewObject():TRUserObject 142
Sample Code (Delphi 5)....142
Sample Code (Visual Basic) 143
SaveToFile(Filename:String) 143
Sample Code (Delphi 5)....143
SelectAll(Option:Integer) 144
Sample Code (Delphi 5)....144
SelectDelete() 144

Sample Code (Delphi 5) 144
SelectPoint(x,y:Double; Option:Integer) 144
Sample Code (Delphi 5) 144
SelectRange(Index, Num:Integer; Option:Integer) 144
Sample Code (Delphi 5) 145
SelectRectangle(X1,Y1:Double; X2,Y2:Double; Option:integer) 145
Sample Code (Delphi 5) 145
Styles[Index:Integer]:ITRStyle 145
Sample Code (Delphi 5) 145
StyleCount:Integer 146
Upper:Double 146
Visible:Boolean 146
16. TSmObject Object 146
Area:Double 147
Sample Code (Visual Basic) 147
Centroid:TRPoint 147
Sample Code (Visual Basic) 147
Counts: array of Integer 147
Sample Code (Visual Basic) 147
Data:String 147
Sample Code (Visual Basic) 147
Fields:Array of string 148
Sample Code (Visual Basic) 148
GetClosestPoint (tp:LongInt; RefPoint:TRPoint; Dist:double):TRPoint 148
Sample Code (Visual Basic) 149
GetPoint(Fraction:Double):TRPoint 149
Sample Code (Visual Basic) 149
ObjectIndex:LongInt 149
Num 149
Sample Code (Visual Basic) 149
Perimeter:Double 150
Sample Code (Visual Basic) 150
Points:Array of TRPoint 150
Sample Code (Visual Basic) 150
ThemeIndex:LongInt 150
Tp:LongInt 150

17.ThemeManager Dialog 151

18.SearchDlg Dialog 156

19.TRBitmap Object 159

Angle:Integer 159

Sample Code (Delphi 5) 159

Handle:Integer 159

Height:Integer 160

LoadImage(FileName:String) 160

Sample Code (Delphi 5) 160

Transparent:Boolean 160

Visible:Boolean 160

Width:Integer 160

20.BackGround Object 161

ColorCenter:Integer 161

ColorDither:Integer 161

Align:Integer 161

Sample Code (Delphi 5) 161

Decay 162

Sample Code (Visual Basic) 162

21.TRViewMgr Object 162

Add(S:TviewRec); 162

Sample Code (Delphi 5) 162

CaptureView(S:String) 163

Sample Code (Delphi 5) 163

Clear() 163

Sample Code (Delphi 5) 163

Count 163

Sample Code (Delphi 5) 163

Delete(N:Integer) 164

Sample Code (Delphi 5) 164

Dialog() 164

Items[Index:integer]: TViewRec 165

Sample Code (Delphi 5) 165

LoadFromFile(S:String) 165

Sample Code (Delphi 5) 165
SaveToFile(S:String) 165
Sample Code (Delphi 5) 165
SetViewByName(S:String) 165
Sample Code (Delphi 5) 165
22. MakeCRA – Archiver Tool 166
23. MakeCRA Operation 166
24. Appendix "A" - Sample .CFG file listing 170
25. Appendix "B" - Sample UserDraw file listing 177
26. Appendix "C" - Sample UserCAD file listing 180
27. Appendix "D" - Sample Theme .TRT file listing 183
28. Appendix "E" - Sample Styles .TRS file listing 188

1. Introduction

MapTivate is a powerful ActiveX component that can be used by developers to embed mapping capabilities into their own products, create stand-alone mapping applications, perform GIS analyses, etc.

Its versatile data import/export engine, make it possible to use with the proprietary, TXF/TDB* vector files, as well as the industry standard MID/MIF and SHP/DBF. It is also capable of using .BMP, .GIF., JPG and .DEM raster image files for greater flexibility.

We recommend that you browse through this document and get an overview of the potential of MapTivate. Then, refer to it when you need help with a specific topic. If you are too busy to go through this document, or to explore all the capabilities in the program, and you need someone to create an application for you quickly, contact Undertow Software to find out about reasonably priced consulting and training services.

2. General Control Description

The MapTivate.OCX control was developed to facilitate mapping and GIS tasks for today's demanding environments. It supports multiple formats, it's easy to use and gives the user a range of capabilities not found in many other products costing thousands of dollars more.

Supported formats

- A number of formats are supported, as shown below. Support for additional formats may be introduced at later dates.
- MID/MIF (MAPINFO data exchange format)
- SHP/DBF (ESRI format)
- TXF/TDB
- CRA Compressed, archived Native format (Created with the **MakeCRA** tool)
- BMP (Windows bitmap format)
- DEM (Digital Elevation Model)
- GIF
- JPG

Map Themes & Styles

Collections of "like" data in one of the above formats, are referred to as Themes. Each theme may contain points, lines, and polygon objects and each object (or groups of objects) in a theme may be assigned one of 256 user-defined styles. Each style, among many other custom properties, may have its own pen, brush, font and mark, as well as visibility thresholds and labeling rules.

Connecting Modes

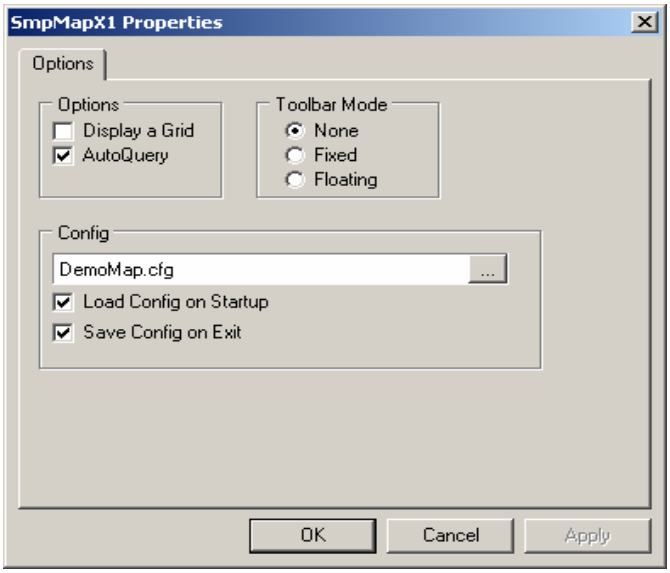
A theme may be ***connected to*** or ***loaded***. If it is connected, then the theme is not loaded, but rather read from the file and played every time the map is redrawn. If it is loaded, then it is literally loaded in RAM and is available for faster map redraws (at the cost of additional RAM requirements).

Installation

MapTivate is a standard OCX. Most developers are familiar with installing and registering OCXes. The MAPTIVATE Installation program copies the control to a location selected by the user and registers it in windows. All the developer needs to do is to register the component in their IDE and then, start a new application and drop an instance of the control on their form. Here is what the control Surface looks like when dropped on a form.

Properties Page

Right clicking on the MapTivate control and then clicking on Properties, or simply double-clicking on the control opens a dialog that allows the user to set the general Control parameters.



Zooming Techniques

Once the application with MapTivate is running, the user may Zoom in or out to see less area in greater detail, or more area in less detail, using a number of different methods. Here is a brief description of these methods (more details are available throughout this document, in the pertinent sections).

Click The Left Mouse Button

Makes the clicked point the new center of the viewport, i.e., the map automatically pans to re-center at the clicked point.

Click Right Mouse Button

Zooms out by a factor of 2.

Click a button on the toolbar (Toolbar needs to be visible)

Depending on which button is pressed you can zoom in, out, to the map extents, etc.

Click and Drag

Clicking the left mouse button and dragging the mouse cursor, while holding the button down, dynamically highlights a rectangular area. When the mouse button is released, the map zooms in so that the highlighted area now fills the viewport.

Record Structures

```
TRPoint = record
        X:double;
        Y:double
    End;

TRExtents = record
        Xmin:double;
        Ymin:double;
        Xmax:double;
        Ymax:double;
     End;

TviewRec = record
        Name:WideString;
        Xref:Double;
        Yref: Double;
        ZoomScale:Double;
      End;

TxObject = record
        Index: LongInt;
        StyleNum: LongInt;
        Points: ^TRPoint;
        Counts: ^LongInt;
Num: LongInt
      End;

TxCADObject = record
Index :LongInt;
StyleNum: LongInt;
Tp: LongInt;
Selected: VARIANT_BOOL;
      End;
```

Enumerations

TxBkPos = set (bkCenter, bkTopLeft, bkTop, bkTopRight, bkRight, bkBottomRight, bkBottom, bkBottomLeft, bkLeft);

TxMouseButton = set (mbLeft, mbRight, mbMiddle);

TxShiftState = set (ssShift, ssAlt, ssCtrl, ssLeft, ssRight);

TxAlignment = set (taLeftJustify, taRightJustify, taCenter,taTop, taVCenter, taBottom)

TxBevelCut = set (bvNone, bvLowered, bvRaised, bvSpace);

TxBorderStyle = set (bsNone, bsSingle);

TxMapMode = set (mmNone, mmZoom, mmDistance);

TxAlignBar = set (asNone, asUpperLeft, asLowerLeft, asUpperRight, asLowerRight);

TxToolbarMode = set (TbNone, TbAnchor, TbFloat);

TxPaintOrder = set (PaintAscending, PaintDescending);

TxConfig = set (cfNone, cfLoad, cfSave, cfLoadSave);

TxPan = set (PanUp, PanUR, PanRight, PanDR, PanDown, PanDL, PanLeft, PanUL)

TxUnits = set (unMi, unKM);

3. MapTivate Object - Properties

AlignBar:TxAlignBar

Defines the position to place the SmpMaX scale bar. One of the following options may be set: asNone, asUpperLeft, asLowerLeft, asUpperLeft, asLowerRight.

Angle:Integer

Angle, in degrees, by which the map is rotated about the current map center. Positive sense is clockwise.

Sample Code (Delphi 5)

```
procedure TForm1.AngleRotClick(Sender: TObject);
begin
```

```
  MapTivate1.angle:=MapTivate1.angle+0.1;
  MapTivate1.RedrawMap;
  panel7.caption:='Rotated(Deg):  '+floattostr(MapTivate1.angle);
end;
```

AutoPaint:Boolean

Sets the behavior of the MapTivate following an operation that changes the viewport. If True, it automatically repaints the map after such operations (ZoomCenter, ZoomHalf, ZoomDouble, ZoomExtents, ZoomPrevious, ZoomSysRect)

Note: Temporarily setting AutoPaint to false permits several time-consuming redraw operations to occur before doing a screen update.

Sample Code (Delphi 5)

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  // Set the Autopaint process to false, so the user
  // can invoke map repainting at will.
  MapTivate1.Autopaint:=false;
  MapTivate1.ZoomCenter(-72,43);
  MapTivate1.smscale:=2;
  MapTivate1.AutoPaint:=true;
  panel1.visible:=false;
end;
```

AutoQuery:Boolean

Sets the behavior of the Automatic querying capabilities of the control. When set to true, as the cursor pointer is moving over the control surface, an OnAutoFindObject event is fired as it moves over an object in an enabled, visible theme. The **msg** variable returned by in the onAutoFindObject event, contains the results according to the QueryStr parameter for the appropriate theme. If multiple themes are loaded, multiple onAutoFindObject events are fired, one for each theme.

A hint is also displayed next to the mouse cursor containing the same information as the **msg** variable in the OnAutoFindObject event for the top-most visible theme.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1AutoFindObject(Sender: TObject; var Theme: ITRTheme;  index:
Integer; const Query: WideString);
begin
    // Make sure Autoquery is true, and this is not caused by a function call
   if MapTivate1.Autoquery=true then
   begin
      ListBox1.items.add('Theme:'+Theme.name+', Object: '+inttostr(index)+'  Info:
'+Query';
   end;
end;
```

Background:IBackground

Access to the Background Object/Interface which allows the user to refine the way the control background is painted prior to the map being rendered. See section on the BackGround Object for details.

BevelInner:TxBevelCut

Sets the Inner Bevel mode for the map control. It can take one of the following enumerated values: bvNone, bvLowered, bvRaised, bvSpace.

BevelOuter:TxBevelCut

Sets the Outer Bevel mode for the map control. It can take one of the following enumerated values: bvNone, bvLowered, bvRaised, bvSpace.

BevelWidth:Integer

Sets the width of the bevel (inner and outer) in pixels.

Sample Code (Delphi 5)

```
procedure TForm1.SetModeClick(Sender: TObject);
begin
  MapTivate1.ActiveMode:=amNone;
  // Set Bevel info
  MapTivate1.bevelinner:=bvraised;
  MapTivate1.bevelouter:=bvlowered;
  MapTivate1.bevelwidth:=4;
end;
```

BorderStyle:TxBorderStyle

Sets the Border style for the Map object. It can take one of the following enumerated values: bsNone, bsSingle.

BorderWidth:Integer

Sets the Border (space between the inner and outer bevels) width in pixels.

BorderColor:Integer

Sets the Color of the border (between the map and the bevel).

Sample Code (Delphi 5)

```
procedure TForm1.SetColsClick(Sender: TObject);
begin
//  Set Border color to blue
  MapTivate1.BorderColor:=clblue;
end;
```

Config:TxConfig

Determines the behavior of the control regarding the configuration file. One of the following values may be selected:

cfNone – Config file is not read at startup and is not written out when exiting the application.
cfLoad - Config file is automatically read during start-up, but is not written out when exiting the application.
cfSave - Config file is not automatically read during start-up, but is written out when exiting the application
cfLoadSave - Config file is automatically read during start-up, and is written out when exiting the application

Sample Code (Delphi 5)

```
procedure TForm1.ConfigClick(Sender: TObject);
const scfg:array[0..3] of string=('cfNone','cfLoad','cfSave','cfLoadSave');
begin
  inc(ncfg);
  if ncfg>3 then ncfg:=0;
  MapTivate1.Config:=TxConfig(ncfg);
  Panel8.caption:='Config:'+scfg[ncfg];
end;
```

Cursor:Integer

Cursor type (Inherited from System). Note that the only time this has any effect is when the Activemode is set to AmNone. Here are some of the available cursor values in Delphi.

crDefault	0
crNone	-1
crArrow	-2
crCross	-3
crIBeam	-4
crSizeNESW	-6
crSizeNS	-7
crSizeNWSE	-8
crSizeWE	-9
crUpArrow	-10
crHourGlass	-11

Enabled:Boolean

Specifies in the Map object will be enabled, i.e. accept mouse events, or not.

Extents:TRExtents

A Read-Only record containing the extents of the current map definition (all themes).

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
var MinX,Miny, MaxX, MaxY:double;
begin
  MinX:=MapTivate1.Extents.Xmin;
  MinY:=MapTivate1.Extents.Ymin;
  MaxX:=MapTivate1.Extents.Xmax;
  MaxY:=MapTivate1.Extents.Ymax;
end;
```

FileName:String

Configuration file name for MapTivate. The extension is .CFG and the default name is **map.cfg.** This configuration file contains the complete definition of the map, e.g., general map settings, themes, styles,…

Grid:Boolean

Determines whether the Gridlines will be displayed on the map, or not.

Sample Code (Delphi 5)

```
procedure TForm1.GridLinesClick(Sender: TObject);
begin
  // Toggle display of grid lines
  MapTivate1.Grid:=not(MapTivate1.Grid);
end;
```

Group:Integer

A convenient way to group Themes for operations that may involve more than a single theme, e.g., when using the FindClosestObject methods.

LastGeo:string

The string that was last used by the search dialog, SearchDlg. The user may set this prior to calling the search dialog to set what appears in the search dialog edit box, when the dialog is opened.

Sample Code (Delphi 5)

```
procedure TForm1.Button32Click(Sender: TObject);
```

```
begin
  with MapTivate1 do
  begin
    // Set string to search for with dialog
    LastGeo:='100 Summer Street,Boston,MA'
    SearchDlg;
  end;
end;
```

MapCenterX:Double

The Longitude of the centerpoint of the viewport. (Read Only Property)

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint2Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Rectangle(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.Point(MapCenterX-1,MapCenterY+1);
    RefreshMap;
  end;
end;
```

MapCenterY:Double

The Latitude of the centerpoint of the viewport. (Read Only Property)

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint2Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Rectangle(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.Point(MapCenterX-1,MapCenterY+1);
    RefreshMap;
  end;
end;
```

MapBottom:Double

The Latitude of the bottom of the viewport. (Read Only Property)

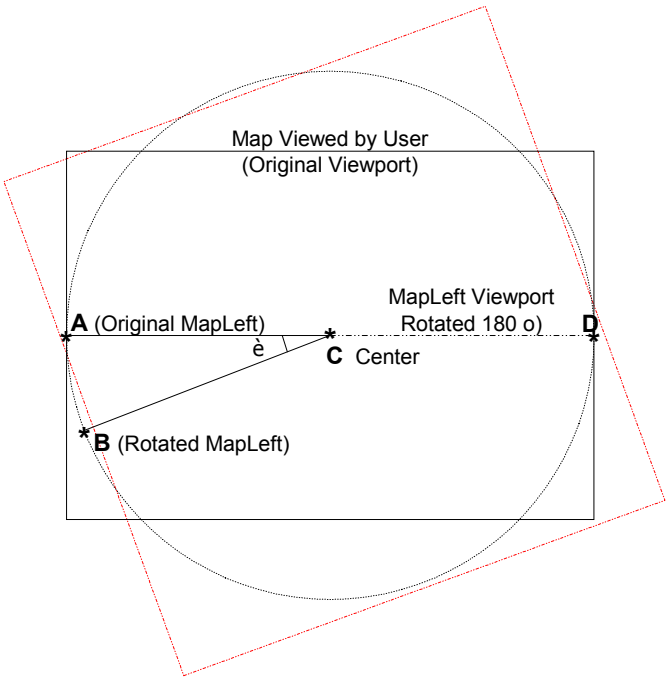
Sample Code (Delphi 5)

```
procedure TForm1.UserPaint2Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Rectangle(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.Point(MapCenterX-1,MapCenterY+1);
    // Change color and width to draw line
```

```
      UserPaint.pen.width:=4;
      UserPaint.pen.color:=clblue;
    UserPaint.Line(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.text(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

MapLeft:Double

The Longitude of map at the left edge of the viewport. (Read Only Property). Note that this only makes sense what the viewport is not rotated. If it is rotated, then it is meaningless (See simple diagram below)



Sample Code (Delphi 5)

```
procedure TForm1.UserPaint2Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Rectangle(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.Point(MapCenterX-1,MapCenterY+1);
    RefreshMap;
  end;
end;
```

MapRight:Double

The Longitude of the right edge of the viewport. (Read Only Property)

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint2Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Rectangle(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.Point(MapCenterX-1,MapCenterY+1);
    RefreshMap;
  end;
end;
```

MapTop:Double

The Latitude of the Top of the viewport. (Read Only Property)

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint2Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Rectangle(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.Point(MapCenterX-1,MapCenterY+1);
    UserPaint.Line(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.text(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

MapMode:TxMapMode

Sets the Map Mode. It can take one of the following enumerated values:

mmNone - The user has complete control of all mouse events.

mmZoom - Default mode of the control. While in this mode, the following behavior is supported.

Click The Left Mouse Button

Makes the clicked point the new center of the viewport, i.e., the map automatically pans to re-center at the clicked point.

Click Right Mouse Button

Zooms out by a factor of 2 (same as ZoomDouble)

Click a button on the toolbar (Toolbar needs to be visible)

Depending on which button is pressed you can zoom in, out, to the map extents, etc.

Click and Drag

Clicking the left mouse button and dragging the mouse cursor, while holding the button down, dynamically highlights a rectangular area. When the mouse button is released, the map zooms in so that the highlighted area now fills the viewport.

mmDistance - While in this mode, the control calculates distances between points successively clicked on, and reports them through an onDistance event, and an onMessage event with code 6. The distance is reported as a total distance for all points and a segment distance between the last point and the current mouse position.

Sample Code (Delphi 5)

```
procedure TForm1.ActModeClick(Sender: TObject);
begin
  inc(AmMode);
  if AmMode>5 then AmMode:=1;
  MapTivate1.ActiveMode:=TxActivemode(AmMode);
  ActMode.Caption:='ActMode:'+inttostr(AmMode);
end;
```

MapName:String

The name of the map, i.e., the collection of themes. This is the string that appear at the top of the themes list in the ThemeMgr dialog.

Sample Code (Delphi 5)

```
procedure TForm1.SetMapNameClick(Sender: TObject);
begin
  // Set MapName and then check through ThemeMgr
  MapTivate1.MapName:='MyMap Title...';
end;
```

MaxZoomScale:Double

The maximum scale value allowed by any of the Zoom operation. The built-in MaxScale corresponds to a horizontal span of 360 degrees.

Sample Code (Delphi 5)

```
procedure TForm1.MxScaleClick(Sender: TObject);
begin
  // Limit zoom out scale to 20 miles
  MapTivate1.MaxZoomScale:=20;
end;
```

MinZoomScale:Double

The minimum scale value allowed by any of the Zoom operation. The built-in MinScale corresponds to 1E-06 of a degree per pixel resolution.

Sample Code (Delphi 5)

```
procedure TForm1.MnScaleClick(Sender: TObject);
begin
  // Limit Zoom in scale to 1 mile
  MapTivate1.MinZoomScale:=1;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command36_Click()
' Set the minimum and maximum Zoom thresholds
With MapTivate1
   ' Permit zom ONLY within 1<x<100 miles
  .Units = 0
  .MinZoomScale = 1
  .MaxZoomScale = 100
  .ZoomExtents
End With
End Sub
```

OverExtend:Integer

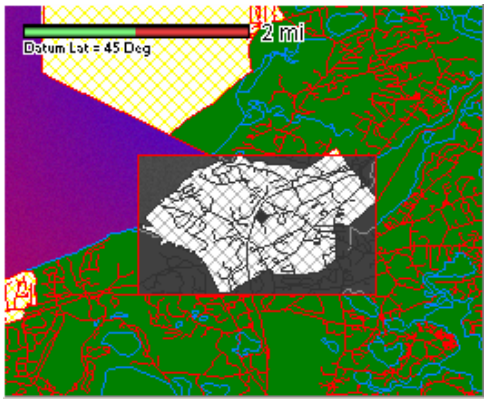
Sets the percent (1 – 100) by which the Zoom operations will oversize the viewport. A value of zero indicates no oversizing.

Sample Code (Delphi 5)

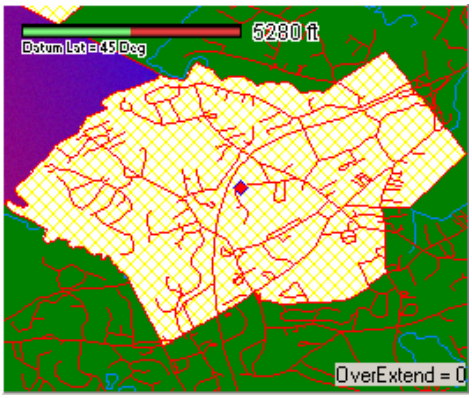
```
procedure TForm1.OverXtndClick(Sender: TObject);
begin
  // Extend the ZoomExtents Area by 50%
  MapTivate1.OverExtend:=50;
end;
```

Zooming into an area with OverExtend=0

Highlighting Area



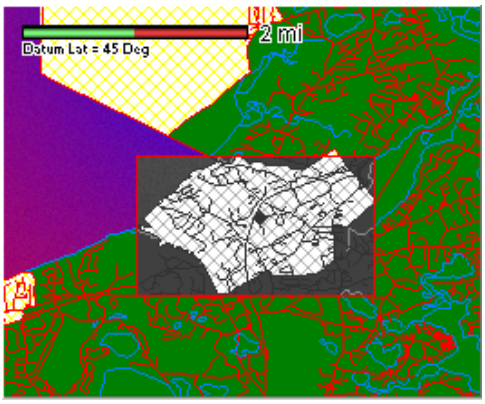
After Redrawing is Complete

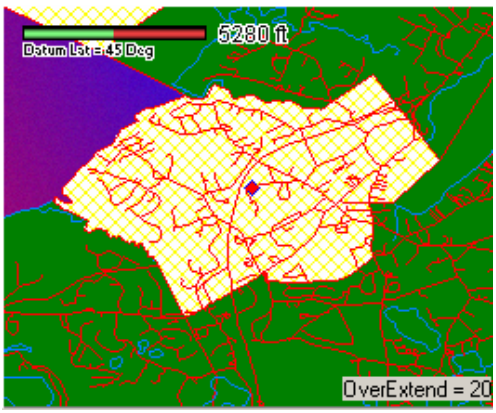


Zooming into an area with OverExtend=20 (20%)

Highlighting Area

After Redrawing is Complete





PaintOrder:TxPaintOrder

Controls the order in which the connected themes are painted on the control's surface. The options are:

PaintAscending

From Themes[0] to Themes[n-1], or top-down in the list of themes shown in ThemeMgr dialog. This is the default setting.

PaintDescending

From Themes[n-1] to Themes[0], or Bottom-up in the list of themes shown in ThemeMgr dialog.

Sample Code (Delphi 5)

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  // Toggle the painting order of the themes
  if MapTivate1.PaintOrder=PaintDescending
       then mpMapX1.PaintOrder:=PaintAscending
            else MapTivate1.PaintOrder:=PaintDescending;
end;
```

PixelSIze:Double

The size of a screen pixel in degrees, at the current scale (based on the center of the viewport). Read Only Property. Used in combination with **Tolerance**.

Sample Code (Delphi 5)

```
procedure TForm1.PixSizeClick(Sender: TObject);
begin
  // Display pixel size in current units
  Panel8.Caption:='Pixel Size:'+floattostr(MapTivate1.pixelsize);
end;
```

RestoreView:Boolean

Determines what happens when an application using MapTivate is started, or any time a configuration file is loaded. If Set to true, then as soon as the configuration file is loaded, the last view setting, stored in the config file, is restored. Otherwise, the control zooms to the extents of the currently loaded theme set. This property may be set in the object inspector (at design time), or may be saved to and loaded from the configuration file.

Sample Code (Delphi 5)

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  MapTivate1.restoreview:=true;
  panel1.visible:=false;
end;
```

RunLicense:Boolean

When true indicates wether the application has a run time license for the executable that was created with the ocx.

ScaleDatum:Double

Sets the latitude to be used to calculate the scale bar displayed on the map. The default value is 45 degrees, which is common for a USA-based map (based on the common –100,45 datum used for Lambert Azimuthal Equal Area Projection for the USA). It should be noted that if a datum other than 45 degrees is used, the displayed maps may not be to scale (the datum was implemented mostly for future use, when other projections may be accommodated by the control).

Sample Code (Delphi 5)

```
procedure TForm1.DatumClick(Sender: TObject);
begin
  // Set Scale datum to 22 degrees
  MapTivate1.ScaleDatum:=22.;
end;
```

ShowProgress:Boolean

If false, disables the display of the "Loading…" message in the middle of the viewport (usually displayed while themes are being loaded).

ThemeCount:Integer;

The number of themes currently connected and/or loaded.

Sample Code (Delphi 5)

```
procedure TForm1.DeleteThemeClick(Sender: TObject);
var n:integer;
begin
  n:=MapTivate1.ThemeCount;
  panel2.caption:='Before:'+inttostr(n);
  // Delete First theme in theme list (remember, zero based
  MapTivate1.DeleteTheme(MapTivate1.themes[0]);
  n:=MapTivate1.ThemeCount;
  panel2.caption:=panel2.caption+',  After:'+inttostr(n);
end;
```

Themes[v:Variant]:ITRTheme

Indexed array referencing the current loaded Themes. Note that **v** is a variant that can be either the index of the theme (integer) or the name of the theme (string).

Sample Code (Delphi 5)

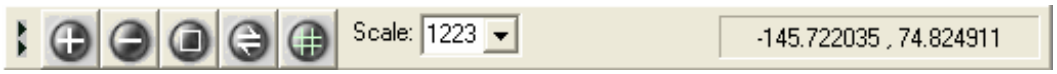
```
procedure TForm1.DeleteThemeClick(Sender: TObject);
var n:integer;
begin
  n:=MapTivate1.ThemeCount;
  panel2.caption:='Before:'+inttostr(n);
  // Delete First theme in theme list (remember, zero based
  MapTivate1.DeleteTheme(MapTivate1.themes[0]);
  n:=MapTivate1.ThemeCount;
  panel2.caption:=panel2.caption+',  After:'+inttostr(n);
end;
```

Tolerance:Integer

The tolerance rectangle (pixels) that is used when selecting items in the viewport using the Autoquery, or the FindObject method. Tolerance is 8 pixels.

ToolBarMode:TxToolBarMode

Set the Tmode of the ToolBar. One of the following enumerated values may be assigned: TbNone, TbAnchor, TbFloat.



TbNone – Cancels the Toolbar
TbAnchor – Anchors the Toolbar on the top left corner of the control.
TbFloat – Floats the Toolbar on the control and allows the user to click on it and drag it to a new location.

Here is an explanation of the various buttons on the ToolBar:

Anchors/Unanchors the Toolbar

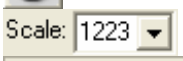

Zoom-In

Zoom-Out

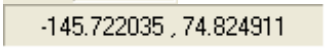
Zoom Extents

Restore Last View

Toggle Grid On/Off



Current scale



Coordinates of current mouse position

Units:TxUnit

Sets the type of units to be used for the map. ***UnMi*** sets the units to miles and *UnKm* sets the units to kilometers.

Sample Code (Delphi 5)

```
procedure TForm1.ScaleClick(Sender: TObject);
begin
  // Set the units to km
  MapTivate1.units:=unkm;
  //Set the scale to 100 miles per virtual display unit
  MapTivate1.smscale:=100;
  // Display ScaleBAr
  MapTivate1.AlignBar:=asUpperLeft;
end;
```

UserCAD:ITRCAD

Interface to the UserCAD object. See the UserCAD Object section for more details.

UserDraw:ITRDraw

Interface to the User Itel theme, to manipulate populations of user bitmaps on the map. See ITRDraw Object section for details.

UserPaint:ITRPaint

Interface to the UserPaint Object that allows the user to draw on the Map surface (see section on ITRPaint Ojbect for detail)

ViewExtents:TRExtents

Current Extents of the Viewport.

Sample Code (Delphi 5)

```
procedure TForm1.VExtentsClick(Sender: TObject);
```

```
begin
  // Display current viewport Extents
  Label2.caption:=floattostr(MapTivate1.ViewExtents.xmin);
  Label3.caption:=floattostr(MapTivate1.ViewExtents.ymin);
  Label4.caption:=floattostr(MapTivate1.ViewExtents.xmax);
  Label5.caption:=floattostr(MapTivate1.ViewExtents.ymax);
end;
```

ViewMrg:ITRViewMgr

Interface to the object that manages user-created views. See ViewMgr Object section for details.

Visible:Boolean

Sets the visibility of the Map control.

ZoomPixels:Integer

The number of pixels that the user generated, rubber-band-rectangle, when in mmZoom mode, has to be, before the operation is a "Zoom" rather than a "Pan".

Sample Code (Delphi 5)

```
procedure TForm1.Z1_PrClick(Sender: TObject);
begin
  // Set the Min # of pixels for Zoom to 12
  MapTivate1.ZoomPixels:=12;
End;
```

ZoomScale:Double

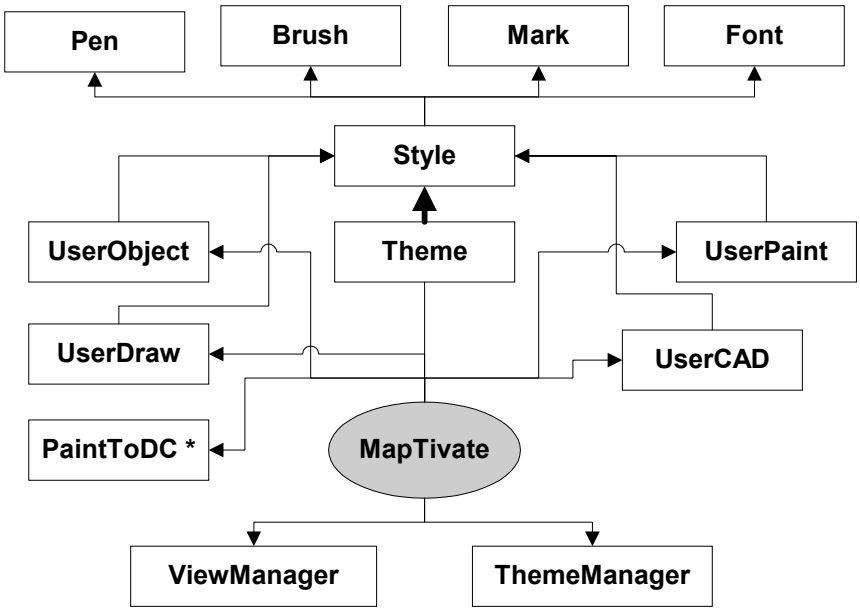
Sets the scale of the map based on the currently selected units and using a virtual 100 pixel scale bar. For example, setting ZoomScale = 50, indicates that 100 pixels of horizontal distance at the current resolution will equal 50 real units (miles or Km, depending on what units the user has selected).

Sample Code (Delphi 5)

```
procedure TForm1.ScaleClick(Sender: TObject);
begin
  // Set the units to km
  MapTivate1.units:=unkm;
  //Set the scale to 100 miles per virtual display unit
  MapTivate1.ZoomScale:=100;
  // Display ScaleBAr
  MapTivate1.AlignBar:=asUpperLeft;
end;
```

4. MapTivate Object - Methods

This is the main map control. All the operations performed using the OCX are mainly managed through this control's interface. Here is a quick diagram of the relationship between the various objects of MapTivate.



** Not an object. Allows drawing map to user-defined DC.*

What follows in the subsequent sections, is a description of all the methods, properties and events of the control, and its objects.

AboutBox()

Displays the version and copyright message of the OCX. This method needs to be available to every application that uses the MapTivate control and is somehow distributed.

*Note that a single user license **does not** entitle you to distribute copies of the MapTivate control. To distribute copies (as part of your application, or in any other form) you need to obtain a special **"Developer's"** license from Undertow Software, Inc.*

Sample Code (Delphi 5)

```
procedure TForm1.AboutClick(Sender: TObject);
begin
  // Display the About Box
  MapTivate1.AboutBox;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command25_Click()
' Display the About Box
   MapTivate1.AboutBox
End Sub
```

ClearMap()

Clears all connected and loaded themes, resets all map properties to their default values, and releases resources back to the pool.

Sample Code (Delphi 5)

```
procedure TForm1.FormCreate(Sender: TObject);
var s:shortstring;
    i:Integer;
begin
  Panel11.Visible:=false;
  // Check to see the status of AutoLoad
  If MapTivate1.Config>0 then
  begin
    s:='AutoLoad Flag is set to: Load or Load & Save.'+#13+#10
         +'This may result in Problems, if files you have connected'
         +'to have been moved, etc.'+#13+#10
         +'Do you want to Cancel loading the Congif file?'#0;
    i:=Application.MessageBox(
    @S[1],
    'Checking the AutoLoad Flag...',
    MB_YESNO);
    // if 6 is returned cancel loading, 7  leave default Mode
    If i=6 then
    begin
      MapTivate1.config:=0;
      MapTivate1.ClearMap;
    end;
    Messagebeep(0);
  end;
  // Set the Autopaint process to false, so the user
  // can invoke map repainting at will.
   MapTivate1.Autopaint:=false;
  // Set restoreview to true
  MapTivate1.restoreview:=true;
  panel1.visible:=false;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
' Clear all map connections and release memory
  MapTivate1.ClearMap
End Sub
```

ClearThemes()

Clears all connected and loaded themes, and releases resources back to the pool, but it does not reset the main map properties, like the ClearMap does.

Sample Code (Delphi 5)

```
procedure TForm1.ClearThemesClick(Sender: TObject);
```

```
begin
  // Clear all themes
  MapTivate1.ClearThemes;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
' Clear all Themes
  MapTivate1.ClearThemes
End Sub
```

ConnectTheme(S:String; Option:LongInt):TRTheme.

Connects the specified Theme to the MapTivate objects and returns a TRTheme interface. Returns nil if the operation fails and no connection is made.

S: String identifying the theme filename. It can be a .MIF, .SHP, .TXF or CRA (native formats), .BMP, .GIF, .JPG, or a .DEM file.

Option: **=1**, Load, **=0** Connect
Specifies whether to connect to the file (and play it every time the map is redrawn, without using any RAM to store the data), or to Load the file in Memory for faster rendering.

It should be obvious that what option is used depends on the particular user application. One is more CPU-intensive, whereas the other is more memory-intensive. For small themes that are redrawn often, **Load** would probably make more sense, if memory is not a problem.

When connecting to an image file such as a BMP, a GIF or a JPG file, the program expects to find a text file with the same name as the image, and the extension **.TIX** in the same folder as the image file. This **.TIX** file contains the coordinates of the lower left and the upper right corner of the bitmap, in degrees, separated by commas, e.g.

-106.737611, 35.001227, -106.529291, 35.209675

These coordinates are used to register (synchronize) the bitmap with the rest of the map themes, and as such it needs to be specified with great care. If the bitmap and the image are not "connected" then the quality of the image will probably suffer as it will be stretched or shrunk to fit the rectangle specified by the coordinates.

Sample Code (Delphi 5)

```
procedure TForm1.ConnectClick(Sender: TObject);
var MyTheme:ITRTheme;
begin
  // Connect to theme and play it when needed to render it
  MyTheme:=MapTivate1.ConnectTheme('mcds.mif',0);
  MessageBeep(0);
  MyTheme.styles[0].Pen.width:=4;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command8_Click()
  ' Check to see if world file is already loaded
  ' If it is, display appropriate message and do not load again
  Set TempCheckTheme = MapTivate1.Themes("World-Countries")
  If ObjPtr(TempCheckTheme) <> 0 Then
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: World-Countries Theme is ALREADY connected!",
vbOKOnly, "Error was encountered!")
  Else
    ' Load Theme from hardwired location
    Set MyWorldTheme = MapTivate1.ConnectTheme("D:\DEVELOP\UNDERTOW\MAPTIVATE\SAMPLE-
SOURCE-DELPHI\country_col_region.shp", 1)
    MyWorldTheme.Name = "World-Countries"
    MapTivate1.ZoomExtents
    WorldIsLoaded = True
  End If
End Sub
```

DeleteTheme(T:TRTheme):LongInt

Deletes the specified theme from the list of connected themes. Note that this does not affect the data file in any way, only the entry in the connected themes list.

Sample Code (Delphi 5)

```
procedure TForm1.DeleteThemeClick(Sender: TObject);
var n:integer;
begin
  n:=MapTivate1.ThemeCount;
  panel2.caption:='Before:'+inttostr(n);
  // Delete First theme in theme list (remember, zero based
  MapTivate1.DeleteTheme(MapTivate1.themes[0]);
  n:=MapTivate1.ThemeCount;
  panel2.caption:=panel2.caption+',  After:'+inttostr(n);
end;
```

Sample Code (Visual Basic)

```
Private Sub Command29_Click()
  ' Check to see if the County Boundary file is already loaded
  Set TempCheckTheme = MapTivate1.Themes("County-Boundary")
  If ObjPtr(TempCheckTheme) <> 0 Then
    ' Delete Theme
    MapTivate1.DeleteTheme TempCheckTheme
    MapTivate1.RedrawMap
  Else
    MsgBox "County Boundary file is NOT Loaded"
  End If
End Sub
```

ExecRegister(n:bstr):Integer

This function is reserved for the developer and controls the registration process of the MapTivate control. **n** controls what happens when the function is called.

n = "0" MapTivate checks the Windows registry to determine if there are any

packages (the OCX, data sets, etc.) that need to be registered with undertow. If any such packages are found, the value of 0 is returned, otherwise, the value of 1 is returned indicating that no such registration is required.

n = "1" MapTivate presents the end user with the appropriate dialogs and information needed to register the product with Undertow Software, Inc.

Sample Code (Visual Basic)

```
Private Sub Command2_Click()
   ' Check to see if any packages need to be registered
   PkNum = MapTivate1.ExecRegister(0)
   Text1.Text = "RegStatus=" & PkNum
   ' Take appropriate action based on result
   If PnNum = 0 Then
   ' Unregistered packages found, call registration dialogs
   MapTivate1.ExecRegister 1
   Else
   End If
End Sub
```

FindClosestObjectClose();

It destroys the list object created by the call to FindFirstClosestObject, and releases all resources used by it.

FindFirstClosestObject(Group:Integer; MaxPoints:LongInt; RefPoint:TRPoint; Distance:Double;CurPoint:TRPoint):TSMObject;

Cycles through all themes in the specified Group, finds the closest points for maximum number of objects specified in MaxPoints, and creates a list object to hold the results. It is normally used as a set with FindNextClosestObject and FindClosestObjectClose.

Group	The Group of Themes to be searched in finding the closest object. (See Theme.Group property) If group 0 is specified, then ALL visible themes are searched, regardless of what group they belong to.
MaxPoints	Maximum number of theme objects to be held on the created list object. The objects are sorted based on their distance from the reference point.
RefPoint	The reference Point used for the ClosestObject calculation
Distance	The distance from the Reference Point to the closest point on the first Returned object.
CurPoint	The coordinates on the closest point on the first object found.

The functions returns a TSMObject which can then be used to retrieve the attributes of the referenced map object. The user may detect when the end of the object list is reached *(if the number of objects found is less than the maximum specified)*, by testing the **.Tp** property of the returned SMObject, to see if it is equal to zero.

Sample Code (Visual Basic)

```
Private Sub Command55_Click()
' Use the FindClosestPoint routines (for ALL themes) for simple Reverse Geocoding
Dim cpt As TRPoint, cptR As TRPoint
Dim Robj As TSmObject
Dim DistX As Double
  List1.Clear
  List1.AddItem "* Starting New Search *"
  DistX = 10 'Distance in Miles, although initialized not used.  A returned value
  MaxNum = 10 'Return only points on the first 10 closest objects
    ' Start Search from Viewport center
  cpt.X = MapTivate1.MapCenterX
  cpt.Y = MapTivate1.MapCenterY
  ' Set Poperties for Marks used to identify closest points later on
  MapTivate1.UserPaint.Mark.Size = 4
  MapTivate1.UserPaint.Mark.Style = 1
  MapTivate1.UserPaint.Mark.Color = vbBlue
  MapTivate1.UserPaint.SmPoint cpt.X, cpt.Y
  MapTivate1.UserPaint.Mark.Size = 12
  MapTivate1.UserPaint.Mark.Style = 0
  cnt = 1 ' Set count to 1 (will be used to label points)
  ' Last theme will be used, make sure one exists.
  If MapTivate1.ThemeCount > 0 Then
    ' Set Group to 0, so ALL themes with default group # will be searched
    Group = 0
    ' Find the closest 10 objects and return the first one
    Set Robj = MapTivate1.FindFirstClosestObject(Group, MaxNum, cpt, DistX, cptR)
    While Robj.Tp <> 0  ' Make sure not at the end of list
      ' List the point, the distance & first field, and visually identify it on the
screen
      List1.AddItem Str(cnt) & ". " & Str(cptR.X) & ", " & Str(cptR.Y) & "[" & Str(DistX)
& "] - " & Robj.Fields(0) & ": " & Str(Robj.ThemeIndex) & " - " & Str(Robj.ObjectIndex)
      MapTivate1.UserPaint.SmPoint cptR.X, cptR.Y
      MapTivate1.UserPaint.SmLine cpt.X, cpt.Y, cptR.X, cptR.Y
      MapTivate1.UserPaint.SmText cptR.X, cptR.Y, Str(cnt)
      ' Get the next closest object
      Set Robj = MapTivate1.FindNextClosestObject(DistX, cptR)
      cnt = cnt + 1
    Wend
    ' Finished.  Close the list object.
    MapTivate1.FindClosestObjectClose
  Else
  End If
  ' Refresh map to show marked points
  MapTivate1.RefreshMap
  Beep
End Sub
```

FindNextClosestObject(Distance:Double;CurPoint:TRPoint):TSMObject;

Finds the next closest Object. It follows a call to FindFirstClosestObject.

Distance	The distance from the Reference Point (see FindFirstClosestObject) to the closest point on the returned object.
CurPoint	The coordinates on the closest point on the first object found.

The functions returns a TSMObject which can then be used to retrieve the attributes of the referenced map object. The user may detect when the end of the object list is reached *(if the number of objects found*

is less than the maximum specified), by testing the **.Tp** property of the returned SMObject, to see if it is equal to zero.

FindGeoFirst(S:String):String;

This is one of a specialized group of functions *(FindGeoFirst, FindGeoNext, FindGeoClose)* that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It creates a list object searches, popsort.txf and returns the first item matching the specified Place, "**S"**. If no matching objects are found, a nul string is returned. In order for this method to work, the files ***popsort.txf*** **and fplcsort.txf,** and the appropriate roads file ***xxR.CRA*** need to be already connected as a themes, prior to calling it. The string S passed to the method may be a address, place name, and a two-letter state abbreviation, separated by a comma.

The specified search string may consist of a Block/Street #, Street Name, City, Place, ZipCode separated by commas. If the block # is omitted, then the search returns all street segments with the specified street name. The ZipCode may be omitted, as well, although that will result is slightly longet search times.

The string that is returned, is also comma-delimited and is composed of the following fields, where available. (Note that the **GeoParse** method may be used to parse these fields if desired).

ADDRESS, STREET, CITY, STATE, ZIP, DISTANCE, QUALITY, COS

Where the fields are explained below.

ADDRESS		Returns the Address (Block) #
STREET		Returns the Name of the street
CITY		Returns the Place or City
STATE		Returns the State (two letter abbreviation)
ZIP		Returns the 5-digit ZipCode
DISTANCE		Returns the distance from the reference point, in miles
QUALITY		Quality factor of the returned result. In detail, it works using the last four bit positions, as follows: bit #4 - City/Place Match (Off - match found, On - match not found) bit #3 - Block Address Match (Off - match found, On - match not found) bits #2 & #1 - Street Match (Both Off - Exact match, bit #1 On - Name and Rode type was matched, bit #2 On - First N characters (N>2) of the main street name matched)
COS		Center-of-Search ID. 1- screen centroid was used, 2- ZipCode centroid was used, 3 - Place centroid was used.

Sample Code (Visual Basic)

```
Private Sub Command47_Click()
  Set TempCheckTheme = MapTivate1.Themes("popsort")
  If ObjPtr(TempCheckTheme) <> 0 Then
    ' Could also Pop Dialog and notify user
    'UserResponse = MsgBox("Problem: PopSort.txf Theme is ALREADY connected!", vbOKOnly,
"Error was encountered!")
  Else
    ' Load Theme from hardwired location
    Set SearchPop =
MapTivate1.ConnectTheme("D:\DEVELOP\UNDERTOW\MAPTIVATE\popplcs\popsort.txf", 1)
  End If
  Set TempCheckTheme = MapTivate1.Themes("r25[MA ROADS]")
  If ObjPtr(TempCheckTheme) <> 0 Then
    ' Could also Pop Dialog and notify user
    'UserResponse = MsgBox("Problem: R25.CRA Theme is ALREADY connected!", vbOKOnly,
"Error was encountered!")
  Else
    ' Load Theme from hardwired location
    Set SearchPlc =
MapTivate1.ConnectTheme("D:\DEVELOP\UNDERTOW\MapTivate\roads\r25.cra", 1)
  End If
  s = MapTivate1.FindGeoFirst("10 ashford lane,andover,ma", MapTivate1.MapCenterX,
MapTivate1.MapCenterY, 10)
  n = MapTivate1.FindGeoClose
  s = MapTivate1.FindGeoFirst("10 ashford lane,andover,ma", MapTivate1.MapCenterX,
MapTivate1.MapCenterY, 10)
  List1.AddItem s
  For i = 1 To n - 1
    s = MapTivate1.FindGeoNext
    List1.AddItem s
  Next i
  n = MapTivate1.FindGeoClose
End Sub
```

FindGeoNext:String;

This is the second of a specialized group of functions *(FindGeoFirst, FindGeoNext, FindGeoClose)* that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It is called after FindGeoFirst and returns the next item in the object list, or nul if no more objects matching the criteria are found. (See Sample code for *FindGeoFirst*).

FindGeoClose:String;

This is the third of a specialized group of functions *(FindGeoFirst, FindGeoNext, FindGeoClose)* that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It is called after all desired calls to FindGeoFirst and FindGeoNext, and it closes the list object and returns all resources to the system pool. (See Sample code for *FindGeoFirst*).

FindObject(X,Y:double);

Searches for all objects within a tolerance (default = 8 pixels) of the specified X,Y coordinates and fires an OnMessage event with code 6, and an OnFindObject event for each element found. The **msg** variable of the OnMessage event contains the information of the found object, formatted using the QueryStr Property for the Theme the found object belongs to. Note that the Enabled property of the theme needs to be set to True for the FindObject to operate as intended.

Sample Code (Delphi 5)

```
procedure TForm1.FindObjClick(Sender: TObject);
Var OXCord,OYCord:double;
       I:integer;
begin
  MapTivate1.AutoQuery:=false;
For I:=0 toMapTivate1.themecount-1 do
Begin
    MapTivate1.Themes[i].Enabled:=true;
    MapTivate1.Themes[i].styles[0].QueryStr:='+F1+F02+F4';
End;
  OXCord:=-70.222674;
  OYCord:=41.655173;
  MapTivate1.smscale:=1;
  MapTivate1.ZoomCenter(OXCord,OYCord);
  MapTivate1.FindObject(OXCord,OYCord);
end;


procedure TForm1.MapTivate1FindObject(Sender: TObject; var Theme: ITRTheme;
  index: Integer; const Query: WideString);
begin
  MessageBeep(0);
  // Event is fired every time a hit is found
  ListBox3.Items.add(Theme.name+':'+inttostr(index)+':'+Query);
end;
```

Sample Code (Visual Basic)

```
Private Sub MapTivate1_Click()
Dim X As Double, Y As Double
  If MapTivate1.MapMode = MmNone Then
    If FileREadyToWrite = True Then
      ' Recording Clicked Points to File
      MapTivate1.GetCoords X, Y
      Npoints = Npoints + 1
      Write #1, X, Y
      ' Mark the point just selected
      MapTivate1.UserPaint.SmPoint X, Y
      MapTivate1.RefreshMap
      ' Annunciate the total # of points captured
      Text1.Text = "Captured: " + Str(Npoints) + " current:" + Str(X) + ", " + Str(Y)
    Else
    End If
    If Calculate = True Then
        MapTivate1.GetCoords X, Y
        MapTivate1.FindObject X, Y
    Else
    End If
  Else
  ' Branch here if mode is NOT mmNone
  End If
End Sub
```

FindPlaceFirst(S:String):String;

This is one of a specialized group of functions *(FindPlaceFirst, FindPlaceNext, FindPlaceClose)* that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It creates a list object searches, popsort.txf and returns the first item matching the specified Place, "**S"**. If no matching objects are found, a nul string is returned. In order for this method to work, the file ***popsort.txf*** needs to be already connected as a theme, prior to calling it. The string S passed to the method may be a place name, or a place name and a two-letter state abbreviation, separated by a comma. A wildcard "*****" is allowed in the Place name, i.e., "**Lowell, MA**", "**L*,MA**", and "**Lawr***" are all valid search strings.

Sample Code (Visual Basic)

```
Private Sub Command46_Click()
  Set TempCheckTheme = MapTivate1.Themes("popsort")
  If ObjPtr(TempCheckTheme) <> 0 Then
    ' Could also Pop Dialog and notify user
    'UserResponse = MsgBox("Problem: PopSort.txf Theme is ALREADY connected!", vbOKOnly,
"Error was encountered!")
  Else
    ' Load Theme from hardwired location
    Set SearchPop =
MapTivate1.ConnectTheme("D:\DEVELOP\UNDERTOW\MAPTIVATE\popplcs\popsort.txf", 1)
  End If
  s = MapTivate1.FindPlaceFirst("Lowe")
  n = MapTivate1.FindPlaceClose
  s = MapTivate1.FindPlaceFirst("Lowe")
  List1.AddItem s
  For i = 1 To n - 1
    s = MapTivate1.FindPlaceNext
    List1.AddItem s
  Next i
  n = MapTivate1.FindPlaceClose
End Sub
```

FindPlaceNext:String;

This is the second of a specialized group of functions *(FindPlaceFirst, FindPlaceNext, FindPlaceClose)* that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It is called after FindPlaceFirst and returns the next item in the object list, or nul if no more objects matching the criteria are found. (See Sample code for *FindPlaceFirst*).

FindPlaceClose:String;

This is the third of a specialized group of functions *(FindPlaceFirst, FindPlaceNext, FindPlaceClose)*that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It is called after all desired calls to FindPlaceFirst and FindPlaceNext, and it closes the list object and returns all resources to the system pool. (See Sample code for *FindPlaceFirst*).

FindZipFirst(S:String):String;

This is one of a specialized group of functions *(FindZipFirst, FindZipNext, FindZipClose)* that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It creates a list object searches, zipsort.txf and returns the first item matching the specified ZipCode, "**S"**. If no matching objects are found, a nul string is returned. In order for this method to work, the file ***zipsort.txf*** needs to be already connected as a theme, prior to calling it.

The string passed to the method can be any valid 5-digit ZipCode, or any partial zipcode followed by an asterisk which acts as a wildcard. Valid search strings are: *01845, 0184**, or even *01**, but not ***845**, i.e., no wildcards on the left allowed.

Sample Code (Visual Basic)

```
Private Sub Command45_Click()
  Set TempCheckTheme = MapTivate1.Themes("zipsort")
  If ObjPtr(TempCheckTheme) <> 0 Then
    ' Could also Pop Dialog and notify user
    'UserResponse = MsgBox("Problem: ZipSort.txf Theme is ALREADY connected!", vbOKOnly,
"Error was encountered!")
  Else
    ' Load Theme from hardwired location
    Set SearchZip =
MapTivate1.ConnectTheme("D:\DEVELOP\UNDERTOW\MAPTIVATE\zipcode\zipsort.txf", 1)
  End If
  s = MapTivate1.FindZipFirst("01810")
  n = MapTivate1.FindZipClose
  s = MapTivate1.FindZipFirst("01810")
  List1.AddItem s
  For i = 1 To n - 1
    s = MapTivate1.FindZipNext
    List1.AddItem s
  Next i
  n = MapTivate1.FindZipClose
End Sub
```

FindZipNext:String;

This is the second of a specialized group of functions *(FindZipFirst, FindZipNext, FindZipClose)* that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It is called after FindZipFirst and returns the next item in the object list, or nul if no more objects matching the criteria are found. (See Sample code for *FindZipFirst*).

FindZipClose:String;

This is the third of a specialized group of functions *(FindZipFirst, FindZipNext, FindZipClose)* that is specifically applicable only to the CRA set of files for the USA, distributed by Undertow Software. It is called after all desired calls to FindZipFirst and FindZipNext, and it closes the list object and returns all resources to the system pool. (See Sample code for *FindZipFirst*).

GeoParse(F:Field; S:String):String;

Parses a string S into fields and returns the value of the specified field, F. Primarily used in parsing a field returned by the GeoFind methods. The ***Field*** specification can actually be a string or an integer and can have the following values:

'ADDRESS' or 1		Returns the Address (Block) #
'NAME' or 'STREET', or 2		Returns the Name of the street
'PLACE' or 'CITY', or 3		Returns the Place or City
'STATE' or 4		Returns the State (two letter abbreviation)
'ZIP' or 'ZIPCODE', or 5		Returns the 5-digit ZipCode
'DISTANCE' or 8		Returns the distance from the reference point, in miles
'QUALITY' or 9		Quality factor of the returned result. In detail, it works using the last four bit positions, as follows: bit #4 - City/Place Match (Off - match found, On - match not found) bit #3 - Block Address Match (Off - match found, On - match not found) bits #2 & #1 - Street Match (Both Off - Exact match, bit #1 On - Name and Rode type was matched, bit #2 On - First N characters (N>2) of the main street name matched)
'COS' or 10		Center-of-Search ID. 1- screen centroid was used, 2- ZipCode centroid was used, 3 - Place centroid was used.

GetArea(P:TXObject):double

Returns the area enclosed by the specified object structure. Note that SmObject.Area should be used, when developing in VB.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1FindObject(Sender: TObject; var Theme: ITRTheme;
  index: Integer; const Query: WideString);
// Display the items found by the query, also the total # found so far.
Var n:integer;
    Area:Double;
    pun,Altun:string;
begin
```

```
   If Calculate=True then
   begin
      n:=index;
      Area:=MapTivate1.GetArea(MyWorldTheme.GetObjectData(n));
      If MapTivate1.Units=unmi then
      begin
        pun:=' sq. mi';
        AltUn:='      ('+FLoattostr(Area*1.609*1.609)+' sq. km)';
      end Else
      begin
        pun:=' sq. km';
        AltUn:='      ('+FLoattostr(Area/1.609/1.609)+' sq. mi)';
      end;
      ListBox3.Items.Add('Theme: '+Theme.Name);
      ListBox3.Items.Add('Country #: '+inttostr(index));
      ListBox3.Items.Add('Area: '+Floattostr(Area)+pun);
      ListBox3.Items.add(AltUn);
      ListBox3.Items.add(' ');
      Panel6.Visible:=true;
      messagebeep(0);
   end;
         end;
```

GetCentroid(P:TXObject):Trpoint;

Returns the Centroid of the specified object structure. Note that SmObject.Centroid should be used, when developing in VB.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1FindObject(Sender: TObject; var Theme: ITRTheme;  index:
Integer; const Query: WideString);
Var n,nn:integer;
    Centroid:TRPoint;
begin
   If Calculate=True then
   begin
     n:=index;
     Centroid:=MapTivate1.GetCentroid(MapTivate1.Themes['World-
Countries'].GetObjectData(n));
     // Mark Centroid Point
     MapTivate1.UserPaint.point(Centroid.x,Centroid.y);
     MapTivate1.RefreshMap;
   end;
end;
```

GetClosestPoint(P:TXObject; T:TRPoint; Dist:Double):TRPoint

Finds a point of the Object **P** that is the closest to the defined point **T**. It also returns the calculated distance,. Note that if **P** is a polygon, and the specified point is inside the polygon, then what is returned by the routine is **T**. Note that SmObject.GetClosestPoint should be used, when developing in VB.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1FindObject(Sender: TObject; var Theme: ITRTheme;  index:
Integer; const Query: WideString);
Var n,nn:integer;
```

```
    Area,Dist:Double;
    Centroid,ClosestF,ClosestT:TRPoint;
    t:TxObject;
begin
   If Calculate=True then
   begin
     n:=index;
     t:=MapTivate1.Themes['World-Countries'].GetObjectData(n);
     if t.points=nil then exit;
    // Find Object Point closest to 0,0
     ClosestF.x:=0;
     ClosestF.y:=0;
     ClosestT:=MapTivate1.GetClosestPoint(t,CLosestF,Dist);
    // Mark closest point
     MapTivate1.UserPaint.point(ClosestT.x,ClosestT.y);
    MapTivate1.RefreshMap;
end;
```

GetCoords(Var x,y:Double)

Returns the coordinates of the position of the mouse cursor, in Lon/Lat.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1Click(Sender: TObject);
Var Points:array[0..N] of TRPoint;
begin
  // Handle the on click event
  If UserClick=true then
  begin
    // Increment point counter
    inc(pctr);
    MapTivate1.GetCoords(Points[pctr].x,Points[pctr].y);
  end;
end;
```

Sample Code (Visual Basic)

```
Private Sub MapTivate1_Click()
Dim X As Double, Y As Double
  If MapTivate1.MapMode = MmNone Then
    If FileREadyToWrite = True Then
      ' Recording Clicked Points to File
      MapTivate1.GetCoords X, Y
      Npoints = Npoints + 1
      Write #1, X, Y
      ' Mark the point just selected
      MapTivate1.UserPaint.SmPoint X, Y
      MapTivate1.RefreshMap
      ' Annunciate the total # of points captured
      Text1.Text = "Captured: " + Str(Npoints) + " current:" + Str(X) + ", " + Str(Y)
    Else
    End If
    If Calculate = True Then
        MapTivate1.GetCoords X, Y
        MapTivate1.FindObject X, Y
    Else
    End If
  Else
  ' Branch here if mode is NOT mmNone
  End If
End Sub
```

GetDistance(X1,Y1,X2,Y2:double):Double;

Calculates the distance between the points specified by X1,Y1 and X2,Y2 in Lon/Lat coordinates, in the currently selected units. Note that this is the Great Circle distance.

Sample Code (Delphi 5)

```
procedure TForm1.GetDistanceClick(Sender: TObject);
Var Dist:double;
begin
  Dist:=MapTivate1.GetDistance(-82,42,-82,41);
  panel3.caption:='Dist: '+floattostr(Dist);
end;
```

Sample Code (Visual Basic)

```
Private Sub Command26_Click()
Dim Point1 As TRPoint, Point2 As TRPoint, Point3 As TRPoint
  ' Find the distance (miles) of one degree Lontitude and Latitude
  Point1.X = -100
  Point1.Y = 44
  '----
  Point2.X = -101
  Point2.Y = 44
  '----
  Point3.X = -100
  Point3.Y = 45
  Dist1 = MapTivate1.GetDistance(Point1.X, Point1.Y, Point2.X, Point2.Y)
  Dist2 = MapTivate1.GetDistance(Point1.X, Point1.Y, Point3.X, Point3.Y)
  Text1.Text = "Calculated Distances = " & Str(Dist1) & "," & Str(Dist2)
End Sub
```

GetPerimeter(p:txobject):double;

Returns the perimeter of the specified object. Note that SmObject.Perimeter should be used, when developing in VB.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1FindObject(Sender: TObject; var Theme: ITRTheme;
  index: Integer; const Query: WideString);
// Display the items found by the query, also the total # found so far.
Var n:integer;
    Perim,Area:Double;
    pun,Altun:string;
begin
   If Calculate=True then
   begin
      n:=index;
      Area:=MapTivate1.GetArea(MyWorldTheme.GetObjectData(n));
     Perim:=MapTivate1.GetPerimeter(MapTivate1.Themes['World-
Countries'].GetObjectData(n));
      If MapTivate1.Units=unmi then
      begin
        pun:=' sq. mi';
        AltUn:='      ('+FLoattostr(Area*1.509*1.509)+' sq. km)';
      end Else
      begin
        pun:=' sq. km';
        AltUn:='      ('+FLoattostr(Area/1.509/1.509)+' sq. mi)';
      end;
      ListBox3.Items.Add('Theme: '+Theme.Name);
      ListBox3.Items.Add('Country #: '+inttostr(index));
```

```
      ListBox3.Items.Add('Area: '+Floattostr(Area)+pun);
      ListBox3.Items.add(AltUn);
     ListBox3.Items.add('Perimeter:' +Floattostr(Perim);
      ListBox3.Items.add(' ');
      Panel6.Visible:=true;
      messagebeep(0);
   end;
         end;
```

GetPoint(P:TXObject,Fraction:double):TRPoint;

Returns the point at a distance of fraction length along the object, e.g. if fraction=0.5 it returns the midpoint. If the specified object is a polygon, then the calculated fraction and point are of the "perimeter" of the object. Note, however, that if the object is composed of multiple polygons, the returned point is meaningless, since all the object perimeters are used in the calculation. Note that SmObject.GetPoint should be used, when developing in VB.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1FindObject(Sender: TObject; var Theme: ITRTheme;   index:
Integer; const Query: WideString);
Var n,nn:integer;
    Perim,Area,Dist:Double;
    Centroid,ClosestF,ClosestT,SomePoint:TRPoint;
    pun,Altun:string;
    t:TxObject;
begin
   If Calculate=True then
   begin
     n:=index;
     t:=MapTivate1.Themes['World-Countries'].GetObjectData(n);
     if t.points=nil then exit;
     // Change the Mark properties so that portions of the polygons can be marked off
     MapTivate1.UserPaint.Mark.color:=clgreen;
     MapTivate1.UserPaint.Mark.Size:=10;
     MapTivate1.UserPaint.Mark.style:=2;
     SomePoint:=MapTivate1.GetPoint(t,0.01);
     MapTivate1.UserPaint.point(SomePoint.x,SomePoint.y);
     MapTivate1.UserPaint.Mark.color:=clyellow;
     SomePoint:=MapTivate1.GetPoint(t,0.1);
     MapTivate1.UserPaint.point(SomePoint.x,SomePoint.y);
     SomePoint:=MapTivate1.GetPoint(t,0.15);
     MapTivate1.UserPaint.point(SomePoint.x,SomePoint.y);
     SomePoint:=MapTivate1.GetPoint(t,0.2);
     MapTivate1.UserPaint.point(SomePoint.x,SomePoint.y);
     MapTivate1.UserPaint.Mark.style:=0;
     MapTivate1.UserPaint.Mark.color:=clblue;
     MapTivate1.UserPaint.Mark.Size:=8;
   end;
end;
```

GotoView(TV:TviewRec)

Sets the viewport to the view specified by TV. It redraws the map after it repositions the ViewPort. The views may be set programmatically, or by invoking the Views Manager Dialog (see description later on).

Sample Code (Delphi 5)

```
procedure TForm1.GoToViewClick(Sender: TObject);
```

```
Var SelectedView:TViewRec;
begin
  // Set View Parameters
  SelectedView.Name:='My Goto View'; // Name of view
  SelectedView.xref:=-85.25;  // Lon
  SelectedView.yref:=-42;   // Lat
  SelectedView.ZoomScale:=1.5;  // Scale in current units
  // Go to Defined View
  MapTivate1.GotoView(SelectedView);
  // Now add it to the Views List, as well
  MapTivate1.ViewMgr.Add(selectedView);
end;
```

Sample Code (Visual Basic)

```
Private Sub Command31_Click()
Dim SelectedView As TViewRec
  Set TempCheckTheme = MapTivate1.Themes("World-Countries")
  If ObjPtr(TempCheckTheme) = 0 Then
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: World-Countries Theme NOT connected!", vbOKOnly,
"Error was encountered!")
  Else
    ' Set View Parameters
    SelectedView.Name = "My VB Greenland View" ' Name of view
    SelectedView.xref = -40.25 ' Lon
    SelectedView.yref = 74 'Lat
    SelectedView.ZoomScale = 500 ' Scale in current units
    ' Go to Defined View
    MapTivate1.GotoView SelectedView
    ' Now add it to the Views List, as well
    MapTivate1.ViewMgr.Add SelectedView
    ' Open the Views Dialog to see it it was added correctly
    MapTivate1.ViewMgr.Dialog
  End If
End Sub
```

HeadsUp(x1,y1,x2,y2:double):LongInt

Rotates the map so the direction “From” X1,Y1 “To” X2,Y2 is upwards. It actually modifies the MapTivate.Angle property. The map direction can be reset by setting the angle to 0. (Note that positive rotation is clockwise).

Sample Code (Delphi 5)

```
procedure TForm1.HeadsUpClick(Sender: TObject);
begin
  // Set heads up by rotating map ~ 45 degrees
  MapTivate1.HeadsUp(-80,40,-81,41);
  MapTivate1.RedrawMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command32_Click()
  'Set heads up by rotating map ~ 45 degrees
  MapTivate1.HeadsUp -80, 40, -81, 41
  MapTivate1.RedrawMap
End Sub
```

LoadProfile(S:String)

Loads the map complete definition from a file specified by the user. The default filename extension is **.cfg** (for configuration). See appendix "A" for a sample printout of a small CFG file. Note that these are flat ASCII (text) files and may be edited using any text editor. Also note that the control expects the string "**MAP:"** *(without the quotes)* at the very beginning of the configuration file. Absence of this string results in an invalid .CFG file.

S: Filename to load the map definition from.

Sample Code (Delphi 5)

```
procedure TForm1.LoadCFGClick(Sender: TObject);
var MyCfgFile:String;
begin
  // Load a previously saved configuration file
  MyCfgFile:='MyFile.CFG';
  MapTivate1.LoadProfile(MyCfgFile);
end;
```

Sample Code (Visual Basic)

```
Private Sub Command33_Click()
Dim MyCfgFile As String
  ' Change the drive and the path to the one where the application is run from
  ChDrive App.Path
  ChDir App.Path
  'Load a configuration file
  MyCfgFile = "MyConfigFile.tmp"
  MapTivate1.LoadProfile MyCfgFile
  MapTivate1.ZoomExtents
End Sub
```

LogOff(FileName:String)

Closes the specified Log File. (See LoGon method)

Sample Code (Visual Basic)

```
Private Sub Command44_Click()
   MapTivate1.LogOff LogFileName
End Sub
```

LogOn(FileName:String; Options:LongInt)

Opens the specified File and starts logging the information generated by the OnMessage event for message codes > 19, mainly for debugging purposes. If Options = 0 then the file is opened as new, clearing any information already in it. If Options = 1, then the file is opened in Append mode and the OnMessage info is added below the information already in the file. When the file is opened, some basic system information is also recorded (Control Build #, memory, screen mode, etc.). Note that this log file is not for the end-user. It is to be created and sent to Undertow Software upon request, when attempting to debug and decipher errant behavior.

Sample Code (Visual Basic)

```
Private Sub Command9_Click()
  ChDrive App.Path
  ChDir App.Path
  LogFileName = Text5.Text
  MapTivate1.LogOn LogFileName, 0
End Sub
```

Sample file generating with the LogOn method…

```
-1:-- SYSTEM INFO --
-1:2/26/2003 11:08:09 PM
-1:Operating System: 5.1
-1:Build Version3.01.00.39
-1:Screen: 1024 x 768 x 24
-1:Processor: Pentium
-1:SYSTEM MEMORY
-1: Total: 535,801 kb
-1: Avail: 168,775 kb
-1:VIRTUAL MEMORY
-1: Total: 2,147,352 kb
-1: Avail: 2,043,498 kb
19:-- Paint Themes --
19:SCALE 3049.588
17:World-Countries
21:Draw Memory:World-Countries
21:Objects=0
21:End of Draw 0.330 (s)
19: Order Labels 0.000 (s)
19:End Paint Themes 0.350 (s)
19:-- Paint Themes --
19:SCALE 3049.588
17:World-Countries
21:Draw Memory:World-Countries
21:Objects=0
21:End of Draw 0.331 (s)
17:USA-States
21:Draw Memory:USA-States
21:Objects=0
21:End of Draw 0.050 (s)
19: Order Labels 0.000 (s)
19:End Paint Themes 0.401 (s)
```

MapToScreen(Xi,Yi:double; Xo,Yo:Integer)

Converts the coordinates of a point from system space (Lat/Lon) coordinates to screen space (pixel) coordinates.

Xi,Yi: System coordinates (Degrees)
Xo,Yo: Screen coordinates (Pixels)

Sample Code (Delphi 5)

```
procedure TForm1.Button4Click(Sender: TObject);
Var X,Y:double;
    Ix,Iy:Integer;
begin
  MapTivate1.MapToScreen(-82,44,ix,iy);
  panel2.caption:='At -82,44: '+inttostr(ix)+', '+inttostr(iy);
end;
```

NewTheme():TRTheme;

Creates a Theme object interface (and adds it to the list of connected themes), that may then be assigned properties, at a later time.

Sample Code (Delphi 5)

```
procedure TForm1.LoadAThemeClick(Sender: TObject);
var MyStyle:TRStyle;
    MyNewTheme:TRTheme;
Begin
  // Create a new Theme interface
  MyNewTheme:=MapTivate1.NewTheme;
  // Load saved theme definition
  MyNewTheme.LoadFromFile('MySingleTheme.TRT');
  MapTivate1.redrawMap;
End;
```

PaintToDC(DC:LongInt; X,Y,W,H:double; Option:Integer);

Paints the currently defined map to the specified DC, using the user specified parameters. Very useful in non-visual environments, when a visible copy of the control can be instantiated.

X, Y: **The offset in reference to the top, left corner of the DC (in pixels)**

W,H: **The width and height of the painted image (in pixels)**

Option: **Options are set based on the bit position in the integer.**

This allows the combination of the various options.

Bit #	*Mode*	*Value*	*Effect*
0	**Not Set**	**0**	**Do NOT Paint the DC canvas background**
0	Set	1	Paint the dc canvas with the control's background color, prior to painting the map on it.
1	**Not Set**	**0**	**Do not scale the map. Clip the map to fit the dc's canvas area, with reference to the top-left corner of the map.**
1	Set	2	Scale the map so that the extents of the map drawn to the dc are the same as those of the main ontrol's surface. The quality of the created map depends on the relative size of the two surfaces.
2	**Not Set**	**0**	**Do not paint a frame around the map.**
2	Set	4	Paint a one-pixel frame around the painted map (using the extents of the current map).

Sample Code (Delphi 5)

```
procedure TForm1.PaintToDCClick(Sender: TObject);
var ndc:integer;
    PaintOpt,w,h:Integer;
begin
   // Default width, height
   w:=205;
   h:=141;
   PaintOpt:=1;
   // Set PaintOpt: 1=Background, 2=Scale
   w:=strtoint(edit13.text);
   h:=strtoint(edit14.text);
   PaintBox1.width:=w;
   PaintBox1.height:=h;
   // Need to process the messages to let windows resize the control
   Application.ProcessMessages;
   PaintOpt:=strtoint(edit12.text);
   // Get dc at this point, dc changes after resize
   ndc:=PaintBox1.Canvas.Handle;
   MapTivate1.PainttoDC(ndc,0,0,w,h,PaintOpt);
end;
```

Sample Code (Visual Basic)

```
Private Sub MapTivate1_PaintAfter(ByVal dc As Long, ByVal w As Long, ByVal h As Long)
  ' Paint to a DC, if the appropriate flag is set
  If PaintToADC.Value = 1 Then MapTivate1.PaintToDC Picture1.hDC, 0, 0,
Picture1.ScaleWidth, Picture1.ScaleHeight, ptDCOptions
End Sub
```

RedrawMap();

Redraws (regenerates) the map bitmap and then paints it on the map surface. Different than RefreshMap which simply repaints the existing bitmap onto the control's surface.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  //Redraw the map from scratch
  MapTivate1.RedrawMap;
end;
```

RefreshMap();

Refreshes the map surface using the existing map bitmap, and flushes any UserPaint operations that result in post processing (e.g., double lines).

Sample Code (Delphi 5)

```
procedure TForm1.Button4Click(Sender: TObject);
begin
  // Refresh map with existing bitmap
  MapTivate1.RefreshMap;
end;
```

SaveToBMP(Filename:String; Format:Integer);

Saves the current map to the specified bitmap (Windows BMP) file. Format controls the format of the created bitmap, and may have one of the following values: 8 – 8bit bitmap, 16 – 16bit bitmap, and 24 – 24bit bitmap.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
var s:String;
begin
  s:='myfile8.bmp';
  //Save Map to 256 color BMP file
  MapTivate1.SaveToBmp(s,8);
  // Load file to verify it was created correctly
  image1.Picture.LoadFromFile('myfile8.bmp');
end;
```

Sample Code (Visual Basic)

```
Private Sub Command20_Click()
'Save the current map image to a bmp file
   CommonDialog1.DialogTitle = "Save map to Bitmap (BMP) File..."
   CommonDialog1.Filter = "Bitmap Files (*.bmp)|*.bmp|All Files|*.*"
   CommonDialog1.ShowSave
   BitMapFile = CommonDialog1.FileName
      MapTivate1.SaveToBmp BitMapFile, 8 ' save to 8-bit BMP
End Sub
```

SavetoGIF(FileName:String);

Saves the current map to the specified GIF file.

Sample Code (Visual Basic)

```
Private Sub Command2_Click()
' Save the current Map image to a GIF file
   CommonDialog1.DialogTitle = "Save map to Bitmap (GIF) File..."
   CommonDialog1.Filter = "GIF Files (*.gif)|*.gif|All Files|*.*"
   CommonDialog1.ShowSave
   BitMapFile = CommonDialog1.FileName
       MapTivate1.SaveToGIF BitMapFile
End Sub
```

SaveToJPG(FileName:String; Format, Quality:Integer)

Saves the current map to the specified JPEG file. Format controls the format of the created bitmap, and Quality the quality of the image. Format may have one of the following values: 8 – 8bit bitmap, and 24 – 24bit bitmap. Quality is an integer from 1 to 100 and indicates the quality due to lossy compression. If 100 is specified, there is no such compression, and more of the original image is preserved. Note that JPG images always have some quality loss due to the nature of the image transformation.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
var s:String;
begin
  s:='myfile8.jpg';
  //Save Map to JPEG file
  MapTivate1.SaveToJPG(s,8,30);
  // Load file to verify it was created correctly
  image1.Picture.LoadFromFile('myfile8.jpg');
end;
```

Sample Code (Visual Basic)

```
Private Sub Command21_Click()
' Save the current Map image to a JPG file
   CommonDialog1.DialogTitle = "Save map to Bitmap (JPG) File..."
   CommonDialog1.Filter = "JPEG Files (*.jpg)|*.jpg|All Files|*.*"
   CommonDialog1.ShowSave
   BitMapFile = CommonDialog1.FileName
       MapTivate1.SaveToJPG BitMapFile, 8, 70
End Sub
```

SaveProfile(S:String)

Saves the complete current map definition to a file specified by the user. The default extension, if not present, is ".CFG". See Appendix "A" for a listing of a sample CFG file.

S: Filename to store the current map definition in.

Sample Code (Delphi 5)

```
procedure TForm1.SaveCFGClick(Sender: TObject);
var MyCfgFile:String;
begin
  // Save the current configuration to a file
  MyCfgFile:='MyFile.CFG';
  MapTivate1.SaveProfile(MyCfgFile);
end;
```

ScreenToMap(Xi,Yi:Integer; Xo,Yo:double)

Converts the coordinates of a point from screen space (pixel) coordinates to System (Lat/Lon) coordinates.

Xi,Yi: Screen coordinates (Pexels)

Xo,Yo: System coordinates (Degrees)

Sample Code (Delphi 5)

```
procedure TForm1.Button4Click(Sender: TObject);
Var X,Y:double;
begin
  // Refresh map with existing bitmap
  MapTivate1.ScreenToMap(100,121,X,Y);
  panel2.caption:='At 100,121: '+floattostr(x)+', '+floattostr(y)+' Degrees';
end;
```

SearchDlg()

Opens the search dialog that permits the user to search for streets, places, Zip Codes, etc. inside the CRA files. See the *SearchDlg Dialog* interface section.

ThemeToBottom(N:Integer):Integer;

Moves the Specified Theme to the Bottom of the paint order and the top of the list order in the Themes Manager dialog. The painting order of the themes may be reversed using the ***PaintOrder*** property. Normally the paint order is Top-to-Bottom (Paintorder=poAscending).

Sample Code (Delphi 5)

```
procedure TForm1.TBotClick(Sender: TObject);
var i,n:integer;
begin
  // Set n, so that no move if theme not found
  n:=-1;
  // Identify Theme in Themelist
  for i:=0 to MapTivate1.ThemeCount-1 do
  begin
    if MapTivate1.themes[i].name='My Theme' then n:=i;
  end;
  // Now, move it to TOP the list
  if n<>-1 then MapTivate1.ThemeToBottom(n);
end;
```

ThemeDown(Index:Integer):LongInt;

Moves the specified Theme one position DOWN (towards the Bottom) in the paint order and the list order in the Themes Manager Dialog. Note that the painting order of the themes may be reversed using the ***PaintOrder*** property.

Sample Code (Delphi 5)

```
procedure TForm1.TDnClick(Sender: TObject);
var i,n:integer;
begin
  // Set n, so that no move if theme not found
  n:=-1;
  // Identify Theme in Themelist
  for i:=0 to MapTivate1.ThemeCount-1 do
  begin
    if MapTivate1.themes[i].name='My Special Theme' then n:=i;
  end;
  // Now, move it one position Down the list
  if n<>-1 then MapTivate1.ThemeDown(n);
end;
```

ThemeToTop(Index:Integer):Integer

Moves the Specified Theme to the Top of the paint order and the top of the list order in the Themes Manager dialog.

Sample Code (Delphi 5)

```
procedure TForm1.TTopClick(Sender: TObject);
var i,n:integer;
begin
  // Set n, so that no move if theme not found
  n:=-1;
  // Identify Theme in Themelist
  for i:=0 to MapTivate1.ThemeCount-1 do
  begin
    if MapTivate1.themes[i].name='My Theme' then n:=i;
  end;
  // Now, move it to TOP of the list
  if n<>-1 then MapTivate1.ThemeToTop(n);
end;
```

ThemeUp(Index:Integer):LongInt;

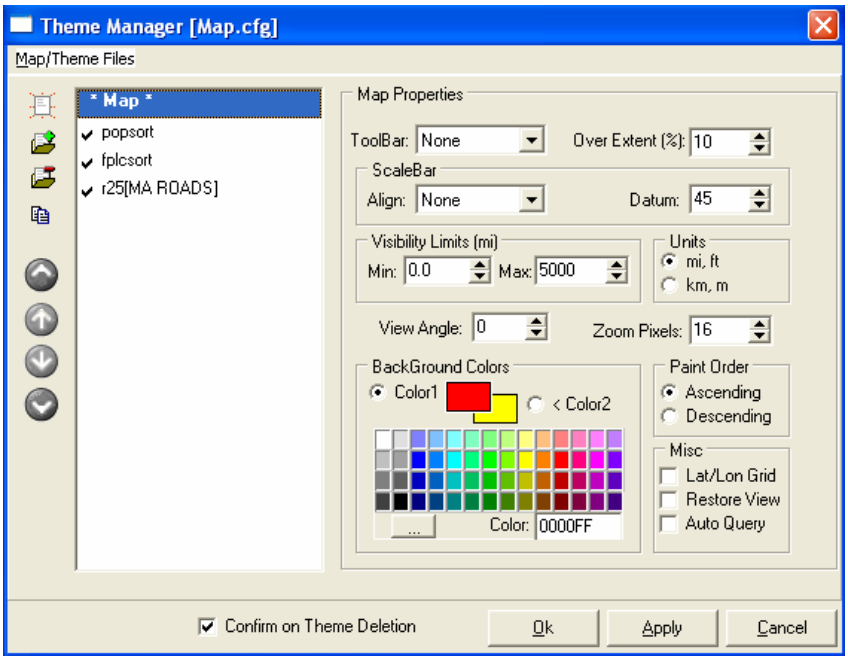
Moves the specified Theme one position UP in the paint order and the list order in the Themes Manager Dialog.

Sample Code (Delphi 5)

```
procedure TForm1.TUpClick(Sender: TObject);
var i,n:integer;
begin
  n:=1;
  // Identify Theme in Themelist
  for i:=0 to MapTivate1.ThemeCount-1 do
  begin
    if MapTivate1.themes[i].name='My Special Theme' then n:=i;
  end;
  // Now, move it one position up the list
  if n<>-1 then MapTivate1.ThemeUp(n);
end;
```

ThemeManager()

Opens the MapTivate Themes Manager Dialog. This manager allows the user to connect themes, specify upper and lower visibility limits, pen and brush properties, etc. in an interactive manner. See **ThemeMgr** Section later in this document.



XYString(FormatString:WideString):Str

Returns the current X,Y coordinates formatted as specified in the FormatString specification (Delphi Formatting string). Here is a brief description of the specification:

Format specifiers have the following form:

"%" [index ":"] ["-"] [width] ["." prec] type

A format specifier begins with a **%** character. After the **%** come the following, in this order:

An optional argument index specifier, **[index ":"]**
An optional left justification indicator, **["-"]**
An optional width specifier, **[width]**
An optional precision specifier, **["." prec]**
The conversion type character, **type**

The following table summarizes the possible values for type:

D	Decimal. The argument must be an integer value. The value is converted to a string of decimal digits. If the format string contains a precision specifier, it indicates that the resulting string must contain at least the specified number of digits; if the value has less digits, the resulting string is left-padded with zeros.
u	Unsigned decimal. Similar to 'd' but no sign is output.
e	Scientific. The argument must be a floating-point value. The value is converted to a string of the form "-d.ddd...E+ddd". The resulting string starts with a minus sign if the number is negative. One digit always precedes the decimal point.The total number of digits in the

	resulting string (including the one before the decimal point) is given by the precision specifier in the format string—a default precision of 15 is assumed if no precision specifier is present. The "E" exponent character in the resulting string is always followed by a plus or minus sign and at least three digits.
f	Fixed. The argument must be a floating-point value. The value is converted to a string of the form "-ddd.ddd...". The resulting string starts with a minus sign if the number is negative.The number of digits after the decimal point is given by the precision specifier in the format string—a default of 2 decimal digits is assumed if no precision specifier is present.
g	General. The argument must be a floating-point value. The value is converted to the shortest possible decimal string using fixed or scientific format. The number of significant digits in the resulting string is given by the precision specifier in the format string—a default precision of 15 is assumed if no precision specifier is present.Trailing zeros are removed from the resulting string, and a decimal point appears only if necessary. The resulting string uses fixed point format if the number of digits to the left of the decimal point in the value is less than or equal to the specified precision, and if the value is greater than or equal to 0.00001. Otherwise the resulting string uses scientific format.
n	Number. The argument must be a floating-point value. The value is converted to a string of the form "-d,ddd,ddd.ddd...". The "n" format corresponds to the "f" format, except that the resulting string contains thousand separators.
m	Money. The argument must be a floating-point value. The value is converted to a string that represents a currency amount. The conversion is controlled by the CurrencyString, CurrencyFormat, NegCurrFormat, ThousandSeparator, DecimalSeparator, and CurrencyDecimals global variables, all of which are initialized from the Currency Format in the International section of the Windows Control Panel. If the format string contains a precision specifier, it overrides the value given by the CurrencyDecimals global variable.
p	Pointer. The argument must be a pointer value. The value is converted to an 8 character string that represents the pointers value in hexadecimal.
s	String. The argument must be a character, a string, or a PChar value. The string or character is inserted in place of the format specifier. The precision specifier, if present in the format string, specifies the maximum length of the resulting string. If the argument is a string that is longer than this maximum, the string is truncated.
x	Hexadecimal. The argument must be an integer value. The value is converted to a string of hexadecimal digits. If the format string contains a precision specifier, it indicates that the resulting string must contain at least the specified number of digits; if the value has fewer digits, the resulting string is left-padded with zeros.

Conversion characters may be specified in uppercase as well as in lowercase—both produce the same results.

For all floating-point formats, the actual characters used as decimal and thousand separators are obtained from the DecimalSeparator and ThousandSeparator global variables.

Index, width, and precision specifiers can be specified directly using decimal digit string (for example "%10d"), or indirectly using an asterisk character (for example "%*.*f"). When using an asterisk, the next argument in the argument list (which must be an integer value) becomes the value that is actually used. For example,

Format('%*.*f', [8, 2, 123.456])

is the same as

Format('%8.2f', [123.456]).

A width specifier sets the minimum field width for a conversion. If the resulting string is shorter than the minimum field width, it is padded with blanks to increase the field width. The default is to right-justify the result by adding blanks in front of the value, but if the format specifier contains a left-justification indicator (a "-" character preceding the width specifier), the result is left-justified by adding blanks after the value.

An index specifier sets the current argument list index to the specified value. The index of the first argument in the argument list is 0. Using index specifiers, it is possible to format the same argument multiple times. For example "Format('%d %d %0:d %1:d', [10, 20])" produces the string '10 20 10 20'.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1MouseMove(Sender: TObject; shift:
 TOleEnum; x,  y: Integer; Xcord, Ycord: Double);
begin
   // Display current Lat, Lon coordinates formatted as shown below
  Panel5.caption:='SysCoord XYString:  '
+MapTivate1.XYString('%6.2f,%6.3f')
end;
```

ZoomCenter(X,Y:Double)

Repositions the map so that the specified point is in the center of the viewport (no change in the scale occurs)

X,Y: Longitude and Latitude of the desired new viewport centerpoint.

Sample Code (Delphi 5)

```
procedure TForm1.ZoomCtrClick(Sender: TObject);
begin
  MapTivate1.ZoomCenter(-92,45);
end;
```

Sample Code (Visual Basic)

```
Private Sub MyGoToCoords_Click()
  ' Goto a specific Lat/Lon, but first make sure map is NOT updated multiple times
  MapTivate1.AutoPaint = False
  MapTivate1.ZoomCenter -82, 44
  ' Set the scale to 500 km
  MapTivate1.ZoomScale = 500
```

```
  MapTivate1.AutoPaint = True
  MapTivate1.Units = Umkm
  ' Mark the spot
  MapTivate1.RedrawMap
  MapTivate1.UserPaint.Mark.Size = 20
  MapTivate1.UserPaint.SmPoint -82, 44
  MapTivate1.RefreshMap
End Sub
```

ZoomDouble()

Zoom the map out so that twice as much area is visible in the viewport (at half the scale).

Sample Code (Delphi 5)

```
procedure TForm1.Z2Click(Sender: TObject);
begin
  MapTivate1.ZoomDouble;
end;
```

ZoomExtents()

Zooms the map in or out, as needed, so that the extents of ALL connected themes (whose Visible flag is set to true) are visible in the viewport.

Sample Code (Delphi 5)

```
procedure TForm1.ZxClick(Sender: TObject);
begin
  MapTivate1.ZoomExtents;
 end;
```

ZoomHalf()

Zoom the map in so that half as much area is visible in the viewport (at twice the scale).

Sample Code (Delphi 5)

```
procedure TForm1.Z1_2Click(Sender: TObject);
begin
  MapTivate1.ZoomHalf;
end;
```

ZoomPan(Dir:TxPan)

Pans the viewport in the user specified direction, by 50%. (TxPan enumeration constants).

Sample Code (Delphi 5)

```
procedure TForm1.ZoomPanVClick(Sender: TObject);
var x1,y,x2:double;
begin
  // Get Vertical Midpoint Coorndinates
```

```
  with MapTivate1 do
  begin
    x1:=MapLeft;
    y:=(MapBottom+MapTop)/2;
    x2:=MapRight;
   end;
  MapTivate1.ZoomPan(PanUp);
  MapTivate1.UserPaint.pen.color:=clblue;
  MapTivate1.UserPaint.pen.width:=4;
  // Draw Reference line
  MapTivate1.UserPaint.Line(x1,y,x2,y);
  MapTivate1.RefreshMap;
end;
```

ZoomPrevious()

Restores the previous map bitmap in the viewport by swapping the current view with the previous one.

Sample Code (Delphi 5)

```
procedure TForm1.Z1_PrClick(Sender: TObject);
begin
  // Restore previous view
  MapTivate1.ZoomPrevious;
 end;
```

ZoomScrnRect(X1,Y1,X2,Y2:Integer);

Zooms the map in or out, as needed, so that the area currently enclosed by the rectangle specified by the two points with screen coordinates X1,Y1 and X2,Y2 fills the viewport.

Sample Code (Delphi 5)

```
procedure TForm1.ZScrClick(Sender: TObject);
begin
  // Zoom the viewport to display what is currently
  // enclosed by the top left, 100x100 pixel rectangle
  MapTivate1.ZoomScrnRect(1,1,100,100);
end;
```

ZoomMapRect(X1,Y1,X2,Y2:double)

Zooms the map in or out, as needed, so that the area enclosed by the rectangle specified by the two points with system coordinates X1,Y1 and X2,Y2 fills the viewport.

Sample Code (Delphi 5)

```
procedure TForm1.ZSysClick(Sender: TObject);
begin
  // Zoom the viewport to display the area
  // enclosed by the rectangle -70,41,-76,44 degrees
  MapTivate1.ZoomMapRect(-70,41,-76,44);
end;
```

ZoomTheme(T:ITRTheme)

Zoom the viewport to the extents of the specified Theme. Useful, for example, if you have loaded all state outlines, each as a different theme, and you only want to zoom so that a "specific" state is visible.

Sample Code (Delphi 5)

```
procedure TForm1.ZThemeClick(Sender: TObject);
begin
  // Zoom to the extents of Theme #1
  If MapTivate1.Themes[1]M< NIL then
         MapTivate1.ZoomTheme(MapTivate1.Themes[1]);
end;
```

5. MapTivate Object - Events

OnAutoFindObject(T:TRTheme; Index:LongInt; Query:String)

If AutoQuery is on, then this event is fired for each object detected within 8 pixels from the location of the mouse Pointer. The Theme **(T)**, the object's index **(Index)** and the string defined by QueryStr **(Query)** are returned by the event.

Sample Code (Visual Basic)

```
Private Sub MapTivate1_AutoFindObject(Theme As MapCtl.TRTheme, ByVal index As Long, ByVal
Query As String)
' Display Found items when Autoquery is ON
   List1.AddItem "Info Found: " & Theme.Name & " :# " & Val(index) & " -  " & Query
End Sub
```

OnClick()

If MapMode is set to mmZoom, clicking and releasing the left mouse button invokes the default behavior, and centers the viewport about the selected point. The Updating of the map takes place on the MouseUp event.

If MapMode is set to amNone, then the user may add their own code to handle the event.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1Click(Sender: TObject);
begin
  // Handle the on click event
  If UserClick=true then
  begin
    // Increment point counter
    inc(pctr);
    MapTivate1.GetCoords(Points[pctr].x,Points[pctr].y);
  end;
end;
```

onDblClick()

There is no default behavior assigned to this event. The user is able to capture it and use it in any way they wish.

onDistance(Sender: TObject; Current, Total: Double);

When the MapMode is set to mmDistance, as the mouse moves over the control surface, an onDistance event is fired, and a line segment from the last clicked point to the current mouse pointer location is drawn. The user may restart the distance calculation, and cancel the current one, by clicking on the right mouse button.

Current = Length of current segment.
Total = total Length

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1Distance(Sender: TObject; Current, Total: Double);
begin
  //  Display Current and Total Distance
  Panel6.caption:='OnDistance: Cur='+floattostr(current)+', Tot='+floattostr(Total);
end;
```

OnFindObject (Sender: TObject; var Theme: ITRTheme; index: Integer; const Query: WideString);

This event is fired for each object found when the FindObject method is called.
The following information is returned:

Theme Theme the found object belongs to
Index The index # of the found item
Query The attributes of the found object, as specified through the QueryStr property.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1FindObject(Sender: TObject; var Theme: ITRTheme;
  index: Integer; const Query: WideString);
begin
  MessageBeep(0);
  ListBox3.Items.add(Theme.name+':'+inttostr(index)+':'+Query);
end;
```

OnFindUserObject (Sender: TObject; const value: ITRObject);

This event is fired for each User object found when the FindObject method is called.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1FindUserObject(Sender: TObject;
  const value: ITRObject);
var s:String;
begin
  s:=floattostr(value.x)+', '+floattostr(value.y);
  Listbox3.Items.add('Found Object #'+inttostr(Value.index)+', Located at: '+s);
end;
```

onMessage(Sender: TObject; Code: Integer; const Msg: WideString);

Triggered when an OCX message has been sent. **Code** is the type of message and **msg** contains the related information.

Code	Msg Contents
1	Floating point system coordinates of the current Mouse location. Message issued when OnMouseMove event is triggered.
2	Screen Coordinates of the current mouse location. Message issued when OnMouseMove event is triggered.
3	Miles per logical inch scale value. Message issued when the map is updated. (Usefull in checking scale settings to ensure they are what the developer meant them to be)
4	Distance in miles. (Note that MapMode needs to be set to mmDistance).
5	Status message indicating a theme is being loaded

19…99	A number of messages are issued as the control goes through loading, and rendering the connected themes. The user may log these messages for the purposes of tracking progress, debugging, etc. (See sample code and output below). Furthermore, if the MapTivate.LogOn mode has been called, this information is also logged in the user-specified Log File.

Sample Code (Visual Basic)

```
Private Sub MapTivate1_Message(ByVal code As Long, ByVal msg As String)
' Record info from OnMessage (for debugging purposes. Ignore messages < 5
  If LogMode = True Then
    If code > 5 Then
      List3.AddItem Val(code) & ": " & msg
    Else
    End If
  Else
  End If
End Sub
```

Sample Output in Listbox, from above code…

```
19:-- Paint Themes --
19:SCALE 3049.588
17:World-Countries
21:Draw Memory:World-Countries
21:Objects=0
21:End of Draw 0.330 (s)
19: Order Labels 0.000 (s)
```

```
19:End Paint Themes 0.350 (s)
19:-- Paint Themes --
19:SCALE 3049.588
17:World-Countries
21:Draw Memory:World-Countries
21:Objects=0
21:End of Draw 0.331 (s)
17:USA-States
21:Draw Memory:USA-States
21:Objects=0
21:End of Draw 0.050 (s)
19: Order Labels 0.000 (s)
19:End Paint Themes 0.401 (s)
```

onMouseDown (Sender: TObject; Button:TxMouseButton; shift: TxShiftState; x, y: Integer; Xcord, Ycord: Double);

Triggered when the left Mouse button is down. No default behavior.

OnMouseMove(Sender: TObject; shift: TxShiftState;Nx,Ny:Integer; Rx,Ry:Double)

Triggered when the mouse is moved within the viewport area. The appropriate Message is also trigerred, allowing the user to obtain certain information (like the X,Y coordinates, etc.)

Nx,Ny: Current screen coordinates (pixels)
Rx,Ry: Current System coordinates (Lat/Lon)

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1MouseMove(Sender: TObject; shift: TOleEnum; x,
  y: Integer; Xcord, Ycord: Double);
begin
   Panel5.font.color:=clblack;
  //- Display Screen Coordinates, if desired
  Panel2.caption:='Screen: '+inttostr(x)+', '+inttostr(y);
  //- Display System Coordinates if desired (Lat/Lon)
  Panel3.caption:='Sys: '+floattostr(Xcord)+', '+floattostr(ycord);
 Panel5.caption:='SysCoord XYString: '+MapTivate1.XYString('%6.2f,%6.3f')
end;
```

onMouseUp(Sender: TObject; Button:TxMouseButton; shift: TxShiftState; x, y: Integer; Xcord, Ycord: Double);

Trigerred when the left mouse button is Up. The default action is to recenter the viewport (see onClick event).

OnPaintBefore(Sender: TObject; dc, w, h: Integer);

Triggered right before the main map is painted. Very useful if the user wants to paint an underlay or change the background color before painting the map.

OnPaintAfter(Sender: TObject; dc, w, h: Integer);

Triggered right after the main map is painted. Very useful if the user wants to paint objects on the map, for example using DrawLine, DrawText or DrawPoint.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1PaintAfter(Sender: TObject; dc, w, h: Integer);
begin
  MapTivate1.UserPaint.Mark.Size:=10;
MapTivate1.UserPaint.Point(MapTivate1.MapCenterX,MapTivate1.MapCenterY);
  MapTivate1.UserPaint.pen.color:=clred;
  MapTivate1.UserPaint.pen.width:=4;
  MapTivate1.UserPaint.Rectangle(-69,41,-70,42);
  MapTivate1.UserPaint.pen.width:=2;
  MapTivate1.UserPaint.pen.color:=clblack;
  MapTivate1.UserPaint.Line(-69,41,-70,42);
  MapTivate1.UserPaint.Font.height:=-18;
  MapTivate1.UserPaint.Font.color:=clblue;
  MapTivate1.UserPaint.Font.BackColor:=clyellow;
  // Italic, bold shadow
  MapTivate1.UserPaint.Font.style:=5;
  MapTivate1.UserPaint.text(-69,41,'Sample Text');
  MapTivate1.RefreshMap;
end;
```

OnResize()

Triggered when the control is resized, at run time.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1Resize(Sender: TObject);
begin
  // Making sure that the image is redrawn after a resize
  Panel7.caption:='Control Was just Resized!';
  MapTivate1.RedrawMap;
  Messagebeep(0);
end;
```

OnSearch (Sender: TObject; const Theme: ITRTheme; index: Integer; const Data: WideString);

Triggered when an object is found that meets the criteria specified in the Theme.SearchData method. Theme is the Theme interface, Index is the object's record # in the theme, and Data is the complete object record, from the database, comma-delimited.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1Search(Sender: TObject; const Theme: ITRTheme;
```

```
  index: Integer; const Data: WideString);
begin
  MessageBeep(0);
  Listbox1.Items.add(theme.name+', index:'+inttostr(index)+', '+Data);
end;
```

OnThemeList()

Triggered when a theme is added or deleted from the current Theme List.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1ThemeList(Sender: TObject);
var i:integer;
begin
  // IF thems list has been modified, then update listbox
  Listbox2.clear;
  for i:=0 to MapTivate1.ThemeCount-1 do
  begin
    listbox2.items.add(MapTivate1.themes[i].name);
  end;
end;
```

6. ITRTheme Object

This object is used to connect to collections of map data. Themes may be thought of as layers that are drawn one on top of the other to produce a map. Since they are drawn sequentially, the user may control this sequence by changing their order in the Theme selection list (programmatically, or through the ThemeMgr dialog). The drawing order of themes may also be controlled through the use of PaintOrder property.

AddStyle():TRStyle;

Adds a style to the list of defined styles (appends it to the end of the list) and increments StyleCount. Note that a Style needs to be added before its attributes can be modified. The only exception is Style[0], which is the default style and is pre-defined.

Sample Code (Delphi 5)

```
 procedure TForm1.Button3Click(Sender: TObject);
begin
  // Add a new style
  MapTivate1.Themes[0].AddStyle;
  // Now use the style before it for subsequent operations
  MapTivate1.Themes[0].StyleNum:=MapTivate1. Themes[0].StyleCount-1;
       end;
```

ClearStyles()

Clears all currently defined styles in the Theme.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  // Delete Style # 5
  MapTivate1.Themes[0].DeleteStyle(5);
  // Clear Current Styles
  MapTivate1.Themes[0].ClearStyles;
  // Now, re-enumerate the styles
  MapTivate1.Themes[0].EnumerateStyles(4,15);
end;
```

ConvertToBMP(FileName:String)

Converts the current bitmap theme to a BMP and its associated TIX registration file. This is valid ONLY for image themes (.BMP, .GIF, .JPG, .DEM).

Sample Code (Delphi 5)

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  // Convert Loaded JPG file to BMP
  TampaJPGSat.ConvertToBMP('MyTampaBMPSat.bmp')
end;
```

ConvertToTXF(FileName:String)

Converts the current theme data to the native (TXF/TDB) format and saves it to the filename specified by the user. It works on SHP/DBF and MID/MIF pairs of files. The advantage is that TXF files are much smaller since the data is stored in binary format.

Sample Code (Delphi 5)

```
procedure TForm1.ssClick(Sender: TObject);
begin
  // Save theme[0] to native format
  MapTivate1.Themes[0].ConvertToTXF('MyNewTxfFile');
end;
```

Count:Integer

The number of objects in the Theme.

Sample Code (Delphi 5)

```
procedure TForm1.GetThemesClick(Sender: TObject);
var i:integer;
begin
  ListBox1.clear;
  // cycle through the connected themes
  for i:=0 to MapTivate1.ThemeCount-1 do
  begin
    ListBox1.Items.Add('Th# '+inttostr(i+1)+', '
+inttostr(MapTivate1.Themes[i].count)+' obj');
  end;
end;
```

Delete()

Deletes the theme from the current list of connected and/or loaded themes.

Sample Code (Delphi 5)

```
procedure TForm1.DeleteThemeClick(Sender: TObject);
var n:integer;
begin
  Panel2.font.color:=clred;
  n:=MapTivate1.ThemeCount;
  panel2.caption:='Before:'+inttostr(n);
  // Delete First theme in theme list (remember, zero based
  MapTivate1.Themes[0].Delete;
  n:=MapTivate1.ThemeCount;
  panel2.caption:=panel2.caption+',  After:'+inttostr(n);
end;
```

DeleteStyle(N:Integer)

Deletes the specified Style from the Theme.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  // Delete Style # 5
  MapTivate1.Themes[0].DeleteStyle(5);
end;
```

Enabled:Boolean

Defines whether the Theme is enabled (True), or Disabled (False). Note that disabled themes are NOT included in the search options.

EnumerateStyles(Index:Integer; Options:Integer)

Automatically create a number of styles for the current theme., using a user-specified field.

Index - The Field Number to use for the enumeration
Options - Bit-position based options as to what properties to auto-assign in the enumeration. 1-Pens, 2-Brushes, 4-Marks, 8-Fieldname.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  // Delete Style # 5
  MapTivate1.Themes[0].DeleteStyle(5);
  // Clear Current Styles
  MapTivate1.Themes[0].ClearStyles;
  // Now, re-enumerate the styles
  MapTivate1.Themes[0].EnumerateStyles(4,15);
```

```
end;
```

Sample Code (Visual Basic)

```
Dim i As Integer
  ' Initialize FoundIndex
  FoundIndex = -1
  ' Determine which Theme is the World-Countries theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "World-Countries" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
    MapTivate1.Themes(FoundIndex).EnumerateStyles 0, 2
    ' Make sure it is NOT autosaved under the same name and replace the config
    MapTivate1.FileName = "MyTemp.cfg"
    MapTivate1.RedrawMap
  Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: World-Countries Theme is NOT connected!", vbOKOnly,
"Error was encountered!")
  End If
End Sub
```

Extents:ITRExtents;

The extents of the Theme (coordinates of Lower left and upper right corner of bounding rectangle).

Sample Code (Delphi 5)

```
procedure TForm1.Themes0ExtClick(Sender: TObject);
begin
  // Display current viewport Extents
  Label2.caption:=floattostr(MapTivate1.Themes[0].Extents.xmin);
  Label3.caption:=floattostr(MapTivate1.Themes[0].Extents.ymin);
  Label4.caption:=floattostr(MapTivate1.Themes[0].Extents.xmax);
  Label5.caption:=floattostr(MapTivate1.Themes[0].Extents.ymax);
end;
```

FileName:String

The filename of the data set connected to the Theme. (ReadOnly)

Sample Code (Delphi 5)

```
procedure TForm1.GetThemeInfoClick(Sender: TObject);
begin
  // Display the name of connected Theme
  Panel2.caption:=MyRoadsTheme.Name;
  // DIsplay Connected Theme filename
  Panel7.caption:=MyRoadsTheme.Filename;
end;
```

FindClosestObjectClose();

It destroys the list object created by the call to FindFirstClosestObject, and releases all resources used by it.

FindFirstClosestObject(MaxPoints:LongInt; RefPoint:TRPoint; Distance:Double;CurPoint:TRPoint):TSMObject;

Finds the closest points for maximum number of objects specified in MaxPoints, and creates a list object to hold the results. It is normally used as a set with FindNextClosestObject and FindClosestObjectClose.

MaxPoints	Maximum number of theme objects to be held on the created list object. The objects are sorted based on their distance from the reference point.
RefPoint	The reference Point used for the ClosestObject calculation
Distance	The distance from the Reference Point to the closest point on the first Returned object.
CurPoint	The coordinates on the closest point on the first object found.

The functions returns a TSMObject which can then be used to retrieve the attributes of the referenced map object. The user may detect when the end of the object list is reached *(if the number of objects found is less than the maximum specified)*, by testing the **.Tp** property of the returned SMObject, to see if it is equal to zero.

Sample Code (Visual Basic)

```
Private Sub Command54_Click()
' Use the FindClosestObject routines for simple Reverse Geocoding
Dim cpt As TRPoint, cptR As TRPoint
Dim Robj As TSmObject
Dim DistX As Double
  List1.AddItem "* Starting New Search *"
  DistX = 10 'Distance in Miles, although initialized not used.  A returned value
  MaxNum = 10 'Return only points on the first 10 closest objects
    ' Start Search from Viewport center
  cpt.X = MapTivate1.MapCenterX
  cpt.Y = MapTivate1.MapCenterY
  ' Set Poperties for Marks used to identify closest points later on
  MapTivate1.UserPaint.Mark.Size = 4
  MapTivate1.UserPaint.Mark.Style = 1
  MapTivate1.UserPaint.Mark.Color = vbBlue
  MapTivate1.UserPaint.SmPoint cpt.X, cpt.Y
  MapTivate1.UserPaint.Mark.Size = 12
  MapTivate1.UserPaint.Mark.Style = 0
  cnt = 1 ' Set count to 1 (will be used to label points)
  ' Last theme will be used, make sure one exists.
  If MapTivate1.ThemeCount > 0 Then
    ' Find the closest 10 objects and return the first one
    Set Robj = MapTivate1.Themes(MapTivate1.ThemeCount -
1).FindFirstClosestObject(MaxNum, cpt, DistX, cptR)
    While Robj.Tp <> 0  ' Make sure not at the end of list
      ' List the point, the distance & first field, and visually identify it on the
screen
      List1.AddItem Str(cnt) & ". " & Str(cptR.X) & ", " & Str(cptR.Y) & "[" & Str(DistX)
& "] - " & Robj.Fields(0)
      MapTivate1.UserPaint.SmPoint cptR.X, cptR.Y
      MapTivate1.UserPaint.SmLine cpt.X, cpt.Y, cptR.X, cptR.Y
      MapTivate1.UserPaint.SmText cptR.X, cptR.Y, Str(cnt)
      ' Get the next closest object
      Set Robj = MapTivate1.Themes(MapTivate1.ThemeCount -
1).FindNextClosestObject(DistX, cptR)
      cnt = cnt + 1
    Wend
    ' Finished.  Close the list object.
    MapTivate1.Themes(MapTivate1.ThemeCount - 1).FindClosestObjectClose
  Else
  End If
  ' Refresh map to show marked points
  MapTivate1.RefreshMap
```

```
    Beep
End Sub
```

FindNextClosestObject(Distance:Double;CurPoint:TRPoint):TSMObject;

Finds the next closest Object. It follows a call to FindFirstClosestObject.

Distance	The distance from the Reference Point (see FindFirstClosestObject) to the closest point on the returned object.
CurPoint	The coordinates on the closest point on the first object found.

The functions returns a TSMObject which can then be used to retrieve the attributes of the referenced map object. The user may detect when the end of the object list is reached *(if the number of objects found is less than the maximum specified)*, by testing the **.Tp** property of the returned SMObject, to see if it is equal to zero.

GetObjectArea(N:Integer):Double;

Returns the Area of the Nth object in the Theme.

GetObjectCentroid(N:Integer):TRPoint;

Returns the centroid of the Nth object in the Theme.

GetObjectClosestPoint(N:LongInt; tp:LongInt; Pt:TRPoint; Dist:Double):TRPoint;

Returns the closest point from Pt to the Nth object in the Theme. The result depends on the value specified for **tp**.

	Type of Object		
Tp	0 – Point	1-Polyline	2-Polygon
0	Return Closest Vertex	Return closest interpolated point	If specified point is outside the object, return point on the perimeter, if inside, return specified point.
1	Return Closest Vertex	Return Closest Vertex	Return Closest Vertex
2	Return closest interpolated point (Treat object as polyline)	Return closest interpolated point (Treat object as polyline)	Return closest interpolated point (Treat object as polyline)
3	Treat object as polyline.	Treat object as	Treat object as

	If specified point is outside the object, return point on the perimeter, if inside, return specified point.	polyline. If specified point is outside the object, return point on the perimeter, if inside, return specified point.	polyline. If specified point is outside the object, return point on the perimeter, if inside, return specified point.

GetObjectData(N:Integer):TXObject;

Returns a pointed to the object structure of the Nth object in the Theme.

Sample Code (Delphi 5)

```
procedure TForm1.MapTivate1FindObject(Sender: TObject; var Theme: ITRTheme;
  index: Integer; const Query: WideString);
Var n,nn:integer;
    Perim:Double;

begin
   If Calculate=True then
   begin
     n:=index;
     Perim:=MapTivate1.GetPerimeter(MapTivate1.Themes['World-
Countries'].GetObjectData(n));
     ListBox3.items.add('Perimeter: '+floattostr(perim));
   end;
end;
```

GetObjectPerimeter(N:Integer):Double;

Returns the perimeter of the Nth object in the Theme.

GetObjectPoint(N:Integer; Fraction:double):TRPoint;

Returns the point at a distance of **Fraction** length along the object, e.g. if fraction=0.5 it returns the midpoint. If the specified object is a polygon, then the calculated fraction and point are of the "perimeter" of the object. Note, however, that if the object is composed of multiple polygons, the returned point is meaningless, since all the object perimeters are used in the calculation.

GetSmObject(N:Integer):TSmObject;

Returns the properties of the object structure of the Nth object in the Theme, and it allows the user access to additional properties and methods of the Nth map object. Primarily added for users in the VB environment which does not permit liberal use of pointers, required for the GetObjectDataMethod, above. (See TsmObject definition later in this manual)

Sample Code (Visual Basic)

```
Private Sub MapTivate1_FindObject(Theme As SmpMapCtl.TRTheme, ByVal index As Long, ByVal
Query As String)
```

```
' Display the items found by the query, also the total # found so far.
Dim n As Integer, nn As Integer
Dim Perim As Double, Area As Double, Dist As Double
Dim Centroid As TRPoint, ClosestF As TRPoint, ClosestT As TRPoint, SomePoint As TRPoint
Dim pun As String, Altun As String
Dim t As TSmObject, myt As TSmObject
Dim i As Integer, j As Integer, xnums As Integer, jj As Integer
Dim count As Integer

   If Calculate = True Then
     n = index
     ' Once we have N, set a criterion for it to change the color of
     ' the selected polygon, but first a temporary style object is created
     If HaveAddedStyle = False Then
       If MapTivate1.Themes("World-Countries").StyleCount = 1 Then
         MapTivate1.Themes("World-Countries").AddStyle
       Else
         MapTivate1.Themes("World-Countries").InsertStyle (1)
       End If
       X = MapTivate1.Themes("World-Countries").StyleCount
       MapTivate1.Themes("World-Countries").Styles(1).Name = "TempShade"
       HaveAddedStyle = True
     Else
     End If
     MapTivate1.Themes("World-Countries").Styles(1).RuleStr = "@Val(@rec)=" & Str(n)
     MapTivate1.Themes("World-Countries").Styles(1).Brush.Color = pBrushColor
     MapTivate1.Themes("World-Countries").Styles(1).Brush.Style = 7
     MapTivate1.Themes("World-Countries").Styles(1).Brush.Transparent = True
     MapTivate1.Themes("World-Countries").Styles(1).formatstr = "F0"

     MapTivate1.Themes("World-Countries").Styles(1).Font.Style = 1
     ' Center-aligned with centroid
     MapTivate1.Themes("World-Countries").Styles(1).Font.Alignment = 3
     MapTivate1.Themes("World-Countries").Styles(1).Font.Color = vbRed
     MapTivate1.Themes("World-Countries").Styles(1).Font.Height = -12
     MapTivate1.Themes("World-Countries").Styles(1).Font.Style = 17
     MapTivate1.Themes("World-Countries").Styles(1).Font.Upper = 5000
     ' Update map with new shaded country
     MapTivate1.RedrawMap

     ' Get some data about the object (country polygon)
     Set t = MapTivate1.Themes("World-Countries").GetSmObject(n)
     DoEvents
     't = MapTivate1.Themes("World-Countries").GetObjectData'(n)
     If t.Counts(0) <= 0 Then
     Else
       ' Calculate the closest point to 0,0 mark points and draw line to it
       ClosestF.X = 0
       ClosestF.Y = 0
       MapTivate1.UserPaint.Mark.Color = vbRed
       MapTivate1.UserPaint.Mark.Size = 6
       MapTivate1.UserPaint.Mark.Style = 1
       ' New closest as part of object.  Find closest point to 0,0
       ClosestT = t.GetClosestPoint(t.Tp, ClosestF, Dist)
       MapTivate1.UserPaint.SmPoint ClosestT.X, ClosestT.Y
       MapTivate1.UserPaint.SmPoint ClosestF.X, ClosestF.Y
       MapTivate1.UserPaint.SmLine ClosestT.X, ClosestT.Y, ClosestF.X, ClosestF.Y

       ' Change the Mark properties so that portions of the polygons can be marked off
       MapTivate1.UserPaint.Mark.Color = vbGreen
       MapTivate1.UserPaint.Mark.Size = 8
       MapTivate1.UserPaint.Mark.Style = 2
       SomePoint = t.GetPoint(0.01)
       MapTivate1.UserPaint.SmPoint SomePoint.X, SomePoint.Y
       SomePoint = t.GetPoint(0.1)
       MapTivate1.UserPaint.SmPoint SomePoint.X, SomePoint.Y
       SomePoint = t.GetPoint(0.15)
       MapTivate1.UserPaint.SmPoint SomePoint.X, SomePoint.Y
       SomePoint = t.GetPoint(0.2)
       MapTivate1.UserPaint.SmPoint SomePoint.X, SomePoint.Y
```

```
        ' Mark Centroid Point
        MapTivate1.UserPaint.Mark.Style = 0
        MapTivate1.UserPaint.Mark.Color = vbYellow
        MapTivate1.UserPaint.Mark.BorderColor = vbRed
        MapTivate1.UserPaint.Mark.Size = 6
        MapTivate1.UserPaint.SmPoint t.Centroid.X, t.Centroid.Y

        ' Get area and perimeter
        Area = t.Area
        Perim = t.Perimeter
        If MapTivate1.Units = UnMi Then
          pun = " sq. mi"
          Altun = "      (" & Str(Area * 1.509 * 1.509) & " sq. km)"
        Else
          pun = " sq. km"
          Altun = "      (" & Str(Area / 1.509 / 1.509) & " sq. mi)"
        End If
        ' Display Area, perimeter, centroid, etc.
        List1.AddItem "Theme: " & Theme.Name
        List1.AddItem "Object Type:" & t.Tp
        List1.AddItem "No of Points: " & Str(t.Num)
        List1.AddItem "Country #: " & Str(index)
        List1.AddItem "Area: " & Str(Area) & pun
        List1.AddItem Altun
        List1.AddItem "Centroid at:" & Str(Centroid.X) & ", " & Str(Centroid.Y)
        List1.AddItem "Perimeter: " & Str(Perim)
        ' Rfresh the map so that the UserPaint objects can become visible
        MapTivate1.RefreshMap
        List1.AddItem " "
      End If
    Else
    End If
End Sub
```

InsertStyle(N:LongInt);

Inserts a style immediately following the style ‘N’ specified by the user.

LoadFromFile(FileName:String)

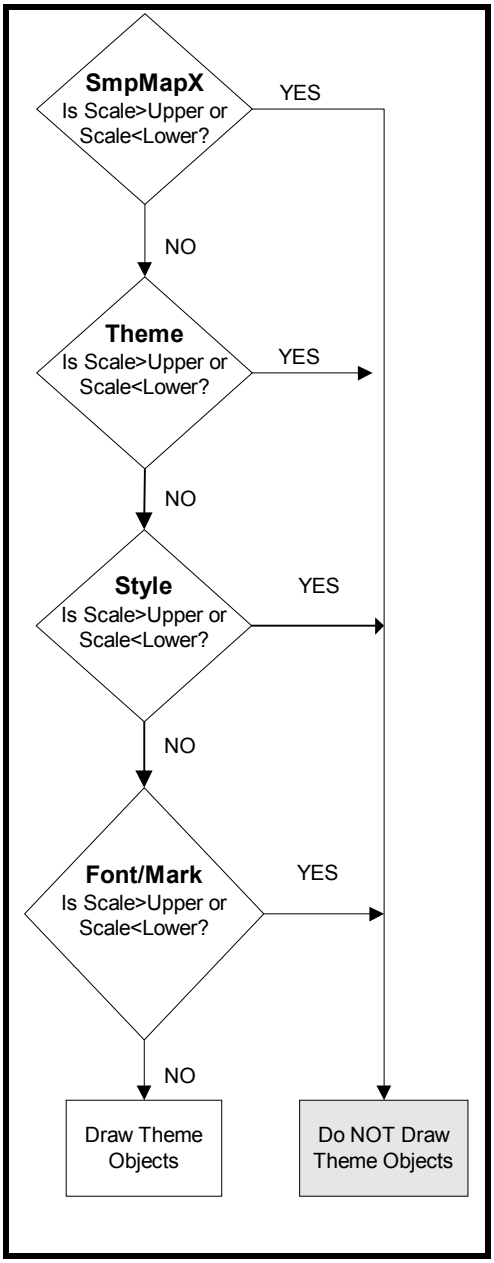
Allows the user to load a single theme configuration file (.TRT). This file contains all the defining properties for the theme, but not the actual vector or data file(s). It simply points to the data files. See Appendix “D” for a sample listing of such a file. (It should be noted that it also contains all the types of information in a style definition .TRS file).

Sample Code (Delphi 5)

```
procedure TForm1.LoadAThemeClick(Sender: TObject);
var MyRoadsTheme:TRTheme;
begin
  //  Connect to Roads Theme
  MyRoadsTheme:=MapTivate1.ConnectTheme('Roads.mif',1);
  // Load Theme Definition File
  MapTivate1.MyRoadsTheme.LoadFromFile('MySingleTheme.TRT');
end;
```

Lower:Double

The lower limit of the scale at which this theme is visible. Remember that the order of precedence for when to make objects visible, is as follows:



Sample Code (Delphi 5)

```
procedure TForm1.LoadAThemeClick(Sender: TObject);
var MyRoadsTheme:TRTheme;
begin
  //  Connect to Roads Theme
  MyRoadsTheme:=MapTivate1.ConnectTheme('Roads.mif',1);
  // Set Upper and lower visibility limits
```

```
  MyRoadsTheme.upper:=30;
  MyRoadsTheme.lower:=0.1;
end;
```

Name:String

The file name the current theme is connected to.

Sample Code (Delphi 5)

```
procedure TForm1.GetThemeInfoClick(Sender: TObject);
begin
  // Display connected Theme Name
  Panel2.caption:=MyRoadsTheme.Name;
end;
```

Priority:Integer

The order of priority in labeling objects. The higher the value, the higher the priority of labeling for this Theme. All labels, for All non-Zero Priority themes are stored and then they are painted based on this theme priority, regardless in the order in which they were connected. A zero value means no priority is assigned to this theme, and the labels are output immediately, with no collision diction processing.

Sample Code (Delphi 5)

```
procedure TForm1.SaveAThemeClick(Sender: Tobject);
begin
   MyUSA.Priority:=5;
  MyRoads.Priority:=2;
  MapTivate1.RedrawMap;
end;
```

SaveToFile(FileName:String)

Saves the current Theme configuration to a .TRT file. See Appendix “D” for a sample listing of a .TRT file.

Sample Code (Delphi 5)

```
procedure TForm1.SaveAThemeClick(Sender: TObject);
begin
  // Save current theme definition to file
  MapTivate1.Themes[0].SaveToFile('MySingleTheme.TRT');
end;
```

SearchData(Ctriterion1, criterion2, Criterion3:String);

Searches the database associated with theme, using up to three criteria. An OnSearch event is fired when an object meeting the criteria is found. Note that less that three criteria may be used by setting the rest of them to a blank string.

Criteria are specified using the rules and evaluator used for FormatStr and RuleStr. For example, searching a world theme with a database where the first field was the country name, then F0=”BOLIVIA” would search the first (zero-based) field until it contained the string BOLIVIA.

Sample Code (Visual Basic)

```
Private Sub Command24_Click()
  FindString = Text3.Text
  Set TempCheckTheme = MapTivate1.Themes("World-Countries")
  If ObjPtr(TempCheckTheme) <> 0 Then
     MapTivate1.Themes("World-Countries").SearchData "+F0='" & FindString & "'", "", ""
  Else
    UserResponse = MsgBox("Problem: World-Countries Theme is NOT Connected!", vbOKOnly,
"Error was encountered!")
  End If
End Sub


Private Sub MapTivate1_Search(ByVal Theme As SmpMapCtl.TRTheme, ByVal index As Long,
ByVal Data As String)
  ' Found the record user is searching for, display whole string
  Beep
  List1.AddItem Data
End Sub
```

StyleCount:Integer

The total number of currently defined styles in the Theme.

Styles[Index:Integer]:ITRStyle

Indexed array that allows the user to access and set the styles to be used for painting the map (See ITRStyle interface). Up to 256 styles are permitted. Styles[0] is the default style.

Sample Code (Delphi 5)

```
procedure TForm1.ConnectClick(Sender: TObject);
var MyTheme:ITRTheme;
begin
  // Connect to theme and load it when needed to render it
  MyTheme:=MapTivate1.ConnectTheme('mcds.mif',0);
  MessageBeep(0);
  MyTheme.styles[0].Pen.width:=4;  // Set Default Pen width
end;
```

Sample Code (Visual Basic)

```
Private Sub Command6_Click()
Dim i As Integer
FoundIndex = -1
' Determine which Theme is the USA-States theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "USA-States" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
  ' Set initial color
  InitColor = &HFF0000
  For i = 0 To MapTivate1.Themes(FoundIndex).StyleCount - 1
    MapTivate1.Themes(FoundIndex).Styles(i).Pen.Width = 1
    MapTivate1.Themes(FoundIndex).Styles(i).Pen.Color = InitColor + 128
```

```
    MapTivate1.Themes(FoundIndex).Styles(i).Pen.OuterWidth = 5
    MapTivate1.Themes(FoundIndex).Styles(i).Pen.Color = InitColor + 128
  Next i
  MapTivate1.RedrawMap
  Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: USA-States Theme is NOT connected!", vbOKOnly, "Error
was encountered!")
  End If
End Sub
```

Upper:Double

The upper scale limit at which this theme is visible. (See description of the property ***Lower*** for precedence in painting the map objects).

Sample Code (Delphi 5)

```
procedure TForm1.LoadAThemeClick(Sender: TObject);
var MyRoadsTheme:TRTheme;
begin
  //  Connect to Roads Theme
  MyRoadsTheme:=MapTivate1.ConnectTheme('Roads.mif',1);
  // Set Upper and lower visibility limits
  MyRoadsTheme.upper:=30;
  MyRoadsTheme.lower:=0.1;
end;
```

Visible:Boolean

Defines the visibility (or not) of the theme. If not visible, the theme does not get included in the ZoomExtents calculation.

7. ITRStyle Object

The TRStyle object contains all the attributes that are needed to customize a Theme. Styles of the theme may also be accessed through and indexed Styles[0..255] array. If no explicit style is specified, then properties set off of TRTheme correspond to Styles[0]. Note that other than Styles[0], you must create a style with AddStyle, before you can use it.

Brush:ITRBrush

Sets the brush properties for this style. See ITRBrush interface.

Sample Code (Delphi 5)

```
procedure TForm1.SetBrushClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
  begin
    // Setting Style for fill brush
    Styles[0].Brush.color:=clyellow;
    Styles[0].Brush.BackColor:=clblue;
```

```
    // Styles[0].Brush.style:=bscross;
  end;
  MapTivate1.RedrawMap;
end;
```

Clear()

Clears current style definition.

Dialog()

Opens the Dialog that allows the user to set all style properties interactively.

Enabled:Boolean

Defines whether this style is enabled, or not.

Font:ITRFont

Sets the font properties for this style. See ITRFont interface for more information.

Sample Code (Delphi 5)

procedure TForm1.NewFontClick(Sender: TObject);

```
begin
  with MapTivate1.Themes[0] do
  begin
    // Setting Font Attributes
    Styles[0].font.name:='verdana';
    Styles[0].font.color:=clblue;
    Styles[0].font.height:=-18;
  end;
  MapTivate1.RedrawMap;
end;
```

FormatStr:String

A string that is used to construct the label to be displayed for each object in the theme, based on the object's attributes. The fields in the attributes are referred to by the key-letter **"F"** followed by the field number, e.g., **F1, F12,** etc.

For example, setting FormatStr to **+F1+"Sample Label"** then the objects would be labeled by the first field in the attribute, plus the string **Sample**.

The following operators are supported in the creating the labels.

Op	Explanation	Example

+	Addition, concatenation	F1+F2=10
-	Subtraction	F11-F10>5
/	Division	(F1/F20)*15<F9
*	Multiplication	(F1*F2)^2<F9
>	Greater Than	F11-F10>5
<	Less that	F1*15<F9
<>	Not Equal to	F9<>200
>=	Equal to or Greater than	F2>=25
<=	Less than or Equal to	F1<=10
^	Raise to power	(F1/F2)^115<F9
=	Equal to	F1+F2=10
( )	Grouping Operation	(F5%3)=2
|	Logical OR	(F2=6)|(F1<5)
&	Logical AND	(F1>15)&(F5=10)
%	Modulo	(F5%3)=2
:n	Format n decimals	

A number of functions are also available in the rule evaluator module, as described below:

@VAL(Str) - Returns the value of Str, where Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n". For example, @VAL(F2) returns the value of the second field of the attribute. If Str is an invalid string then zero is returned.

@CHR(N) - Returns the character with ordinal N (integer). The argument can also be the variable Fn, i.e., Attribute Field "n", e.g. @CHR(12), or @CHR(F1)

@LENGTH(Str) - Returns the length of Str, where Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n", e.g. @LENGTH('Test'), or @LENGTH(F1+F5)

@COPY(Str,n1,n2) - Similar to the Delphi Copy function, where it returns the copy of Str stating at position n1 and n2 characters long. Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n", e.g., @COPY(F2,2,5)

@POS(SubStr,Str) - Returns an integer indicating the position of SubString in Str. Substring and Str can be any valid string, or they can be the variable Fn, i.e., Attribute Field "n", e.g., @POS('e','Sample') would return 6

@REC - Returns the current record number, as a string. @VAL(@REC)) may be used if the value is desired.

@TRIM(Str) – Returns the string stripped of leading and trailing spaces.

@FORMAT(Fst,Value) - Returns the real **value** formatted using the specification Fst. (For details on this type of formatting see the section for the Poperty MapTivate.XYString.

@IF(Criterion,TrueResult,FalseResult) – Branches to the operation defined in TrueResult, or FalseResult based on the evaluation of Criterion.

Sample Code (Delphi 5)

```
procedure TForm1.LoadAThemeClick(Sender: TObject);
begin
  //  Connect to Roads Theme
  MyRoadsTheme:=MapTivate1.ConnectTheme('Roads.mif',1);
  // Set Upper and lower visibility limits
  MyRoadsTheme.upper:=30;
  MyRoadsTheme.lower:=0.1;
  // Set labeling options
  MyRoadsTheme.Styles[0].FormatStr:='Name=+06';
  MyRoadsTheme.Styles[0].font.upper:=1;
  MyRoadsTheme.Styles[0].font.lower:=0.05;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command9_Click()
Dim i As Integer
  FoundIndex = -1
  ' Determine which Theme is the World-Countries theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "County-Boundary" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
  ' Change some of the county theme attributes
  ' General RGB Color setting is &HBBGGRR&
  MyCountyBoundary.Styles(0).Brush.Color = vbYellow  ' Set County color
    Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: County Boundary Theme is NOT connected!", vbOKOnly,
"Error was encountered!")
  End If

  FoundIndex = -1
  ' Determine which Theme is the World-Countries theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "County-Water" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
  MyCountyWater.Styles(0).Pen.Color = vbBlue  ' Set Water color
  MyCountyWater.Styles(0).Brush.Color = vbBlue
      Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: County Water Theme is NOT connected!", vbOKOnly,
"Error was encountered!")
  End If

  FoundIndex = -1
  ' Determine which Theme is the World-Countries theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "County-Roads" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
  ' Set parameters to display Road names at certain scales
    MyCountyRoads.Styles(0).Pen.Color = &H11AABB  ' Set Roads color
    MyCountyRoads.Styles(0).formatstr = "F03"
    MyCountyRoads.Styles(0).Font.Upper = 5
    MyCountyRoads.Styles(0).Font.Lower = 0
    ' Make sure collision detection is ON so labels are readable
    MyCountyRoads.Styles(0).Font.CollisionDetection = True
    ' Label each road in the viewport ONLY once
    MyCountyRoads.Styles(0).Font.UseOnce = True
  Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: County Roads Theme is NOT connected!", vbOKOnly,
"Error was encountered!")
```

```
  End If
  ' redraw, just in case
  MapTivate1.RedrawMap
End Sub
```

ID:Integer

The index on the style in the Styles[] list, 0-based.

Sample Code (Delphi 5)

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  // Simply demonstrated the ID equivalency with the indexed position of Style
  Panel5.caption:=inttostr(MapTivate1.Themes[0].Styles[5].ID);
end;
```

Lower:Double

Lower limit of visibility for the Style. See the beginning of this section for how the upper and lower settings of each object hierarchically affect the visibility of map features.

Mark:ITRMark

Sets the mark properties (marker that is used to identify point objects) for this style. See ITRMark interface for more information.

Sample Code (Delphi 5)

```
procedure TForm1.MarksClick(Sender: TObject);
begin
with MapTivate1.Themes[0] do
  begin
    // Setting Mark Attributes
    // Mark Polygon Centroids
    Styles[0].Mark.VisFilter:=4;
    Styles[0].Mark.Visible:=true;
    Styles[0].Mark.Style:=2;
    Styles[0].Mark.Upper:=20;
    Styles[0].Mark.Lower:=0.1;
    Styles[0].Mark.Color:=clBlue;
    Styles[0].Mark.BorderColor:=clRed;
    Styles[0].Mark.Size:=14;
  end;
  MapTivate1.RedrawMap;
end;
```

Name:String

The name assigned to this style (it may be assigned programmatically, or through the Style dialog, see section further in this document).

Sample Code (Delphi 5)

```
procedure TForm1.GetStyleInfoClick(Sender: TObject);
begin
  // Display Style Name
  Panel3.caption:=MyRoadsTheme.styles[0].Name;
end;
```

Pen:ITRPen

Sets the pen properties for this style. See ITRPen interface for more information.

Sample Code (Delphi 5)

```
procedure TForm1.ConnectClick(Sender: TObject);
var MyTheme:ITRTheme;
begin
  // Connect to theme and load it when needed to render it
  MyTheme:=MapTivate1.ConnectTheme('mcds.mif',0);
  MessageBeep(0);
  MyTheme.styles[0].Pen.width:=4;
end;
```

RuleStr:String

A string that is used to determine what objects will be assigned to this style, based on a calculation involving the contents of the attributes file of the Theme. The fields in the attributes are referred to by the key-letter **"F"** followed by the field number, e.g., **F1, F12,** etc. The following operators are supported in the calculations involving the fields.

Op	Explanation	Example
+	Addition, concatenation	F1+F2=10
-	Subtraction	F11-F10>5
/	Division	(F1/F20)*15<F9
*	Multiplication	(F1*F2)^5<F9
>	Greater Than	F11-F10>5
<	Less that	F1*15<F9
<>	Not Equal to	F9<>200
>=	Equal to or Greater than	F2>=25
<=	Less than or Equal to	F1<=10
^	Raise to power	(F1/F2)^115<F9
=	Equal to	F1+F2=10
( )	Grouping Operation	(F5%3)=2
|	Logical OR	(F2=6)|(F1<5)
&	Logical AND	(F1>15)&(F5=10)
%	Modulo	(F5%3)=2
:n	Format n decimals	

A number of functions are also available in the rule evaluator module, as described below:

@VAL(Str) - Returns the value of Str, where Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n". For example, @VAL(F2) returns the value of the second field of the attribute. If Str is an invalid string then zero is returned.

@CHR(N) - Returns the character with ordinal N (integer). The argument can also be the variable Fn, i.e., Attribute Field "n", e.g. @CHR(12), or @CHR(F1)

@LENGTH(Str) - Returns the length of Str, where Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n", e.g. @LENGTH('Test'), or @LENGTH(F1+F5)

@COPY(Str,n1,n2) - Similar to the Delphi Copy function, where it returns the copy of Str stating at position n1 and n2 characters long. Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n", e.g., @COPY(F2,2,5)

@POS(SubStr,Str) - Returns an integer indicating the position of SubString in Str. Substring and Str can be any valid string, or they can be the variable Fn, i.e., Attribute Field "n", e.g., @POS('e','Sample') would return 6.

@REC - Returns the current record number, as a string.

@TRIM(Str) – Returns the string stripped of leading and trailing spaces.

@FORMAT(Fst,Value) - Returns the real **value** formatted using the specification Fst. (For details on this type of formatting see the section for the Poperty MapTivate.XYString.

@IF(Criterion,TrueResult,FalseResult) – Branches to the operation defined in TrueResult, or FalseResult based on the evaluation of Criterion.

Sample Code (Delphi 5)

```
procedure TForm1.Button8Click(Sender: TObject);
begin
  // Set New rules for Styles 1 and 2
  // Also Set the FormatStrs
  With MapTivate1.Themes[0] do
  begin
    Styles[1].RuleStr:='@Val(F03)>50';
    Styles[1].FormatStr:='+F4';
    Styles[2].RuleStr:='@Val(F03)>5';
    Styles[1].FormatStr:='+"["+F4+"]"';
  end;
end;
```

QueryStr:String

A string that is used to construct the label to be displayed for each object found when the FindObject method is used. Its format is the same as the FormatStr property. The fields in the attributes are referred to by the key-letter **"F"** followed by the field number, e.g., **F1, F12,** etc. For example, setting FormatStr to **+F1+"Sample Label"** then the string that would be returned when the object was queried, would be made up of the first field in the attribute, plus the string **Sample**.

The following operators are supported in the creating the labels.

Op	Explanation	Example
+	Addition, concatenation	F1+F2=10
-	Subtraction	F11-F10>5
/	Division	(F1/F20)*15<F9
*	Multiplication	(F1*F2)^5<F9
>	Greater Than	F11-F10>5
<	Less that	F1*15<F9
<>	Not Equal to	F9<>200
>=	Equal to or Greater than	F2>=25
<=	Less than or Equal to	F1<=10
^	Raise to power	(F1/F2)^115<F9
=	Equal to	F1+F2=10
( )	Grouping Operation	(F5%3)=2
|	Logical OR	(F2=6)|(F1<5)
&	Logical AND	(F1>15)&(F5=10)
%	Modulo	(F5%3)=2
:n	Format n decimals	

A number of functions are also available in the rule evaluator module, as described below:

@VAL(Str) - Returns the value of Str, where Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n". For example, @VAL(F2) returns the value of the second field of the attribute. If Str is an invalid string then zero is returned.

@CHR(N) - Returns the character with ordinal N (integer). The argument can also be the variable Fn, i.e., Attribute Field "n", e.g. @CHR(12), or @CHR(F1)

@LENGTH(Str) - Returns the length of Str, where Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n", e.g. @LENGTH('Test'), or @LENGTH(F1+F5)

@COPY(Str,n1,n2) - Similar to the Delphi Copy function, where it returns the copy of Str stating at position n1 and n2 characters long. Str is any valid string, or it can be the variable Fn, i.e., Attribute Field "n", e.g., @COPY(F2,2,5)

@POS(SubStr,Str) - Returns an integer indicating the position of SubString in Str. Substring and Str can be any valid string, or they can be the variable Fn, i.e., Attribute Field "n", e.g., @POS('e','Sample') would return 6

@REC - Returns the current record number, as a string.

@TRIM(Str) – Returns the string stripped of leading and trailing spaces.

@FORMAT(Fst,Value) - Returns the real **value** formatted using the specification Fst. (For details on this type of formatting see the section for the Poperty MapTivate.XYString.

@IF(Criterion,TrueResult,FalseResult) – Branches to the operation defined in TrueResult, or FalseResult based on the evaluation of Criterion.

Sample Code (Delphi 5)

```
procedure TForm1.FindObjClick(Sender: TObject);
Var OXCord,OYCord:double;
begin
  MapTivate1.AutoQuery:=false;
  MapTivate1.Themes[0].styles[0].QueryStr:='+01';
  OXCord:=-70.222674;
  OYCord:=41.655173;
  MapTivate1.smscale:=1;
  MapTivate1.ZoomCenter(OXCord,OYCord);
  MapTivate1.FindObject(OXCord,OYCord);
end;
```

Upper:Double

Upper limit of visibility for the User Item Theme.

Visible:Boolean

Sets the visibility of this Style

8. ITRPen Object

The Pen object is used to set all pen properties.

BackColor:Integer

Pen background color. This is the color used to draw the line specified by the Outerwidth (see later on). Combining Color, BackColor, Width and OuterWidth the user can also create double-lined roads.

Sample Code (Delphi 5)

```
procedure TForm1.SetPenClick(Sender: TObject);
begin
  MapTivate1.Themes[0].Styles[0].Pen.color:=clblue;
  MapTivate1.Themes[0].Styles[0].Pen.width:=1;
 // MapTivate1.Themes[0].Styles[0].Pen.style:=psdash;
  MapTivate1.Themes[0].Styles[0].Pen.BackColor:=clred;
  MapTivate1.Themes[0].Styles[0].Pen.outerwidth:=12;
  MapTivate1.RedrawMap;
end;
```

Color:Integer

Pen foreground color. This is the color used to draw the line specified by the Width (see later on). Combining Color, BackColor, Width and OuterWidth the user can also create double-lined roads.

Sample Code (Delphi 5)

```
procedure TForm1.SetPenClick(Sender: TObject);
begin
  MapTivate1.Themes[0].Styles[0].Pen.color:=clblue;
  MapTivate1.Themes[0].Styles[0].Pen.width:=1;
 // MapTivate1.Themes[0].Styles[0].Pen.style:=psdash;
  MapTivate1.Themes[0].Styles[0].Pen.BackColor:=clred;
  MapTivate1.Themes[0].Styles[0].Pen.outerwidth:=12;
  MapTivate1.RedrawMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command6_Click()
Dim i As Integer
FoundIndex = -1
' Determine which Theme is the USA-States theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "USA-States" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
  ' Set initial color
  InitColor = &HFF0000
  For i = 0 To MapTivate1.Themes(FoundIndex).StyleCount - 1
    MapTivate1.Themes(FoundIndex).Styles(i).Pen.Width = 1
    MapTivate1.Themes(FoundIndex).Styles(i).Pen.Color = InitColor + 128
    MapTivate1.Themes(FoundIndex).Styles(i).Pen.OuterWidth = 5
    MapTivate1.Themes(FoundIndex).Styles(i).Pen.Color = InitColor + 128
  Next i
  MapTivate1.RedrawMap
  Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: USA-States Theme is NOT connected!", vbOKOnly, "Error
was encountered!")
  End If
End Sub
```

Mode:Integer

Reserved.

OuterWidth:Integer

Allows the user to draw thick lines, or a double line by specifying an Outerwidth wider that the Width property (and preferably a different color). Note that the process is to draw the “OuterWidth” line first and then draw the “Width” line on top of it.

Sample Code (Delphi 5)

```
procedure TForm1.SetPenClick(Sender: TObject);
begin
  MapTivate1.Themes[0].Styles[0].Pen.color:=clblue;
  MapTivate1.Themes[0].Styles[0].Pen.width:=1;
  MapTivate1.Themes[0].Styles[0].Pen.BackColor:=clred;
  MapTivate1.Themes[0].Styles[0].Pen.outerwidth:=12;
  MapTivate1.RedrawMap;
end;
```

Style:Integer;

Pen style. It can take any of the following values:

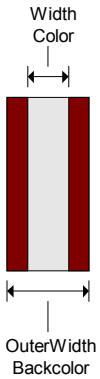
Style Value	Delphi Equivalent	Description
0	**PsSolid**	**A solid line.**
1	**PsDash**	**A line made up of a series of dashes**
2	**PsDot**	**A line made up of a series of dots**
3	**PsDashDot**	**A line made up of alternating dashes and dots**
4	**PsDashDotDot**	**A line made up of a serious of dash-dot-dot combinations**
5	**PsClear**	**No line is drawn (used to omit the line around shapes that draw an outline using the current pen).**
6	**PsInsideFrame**	**A solid line, but one that may use a dithered color if Width is greater than 1.**

Sample Code (Delphi 5)

```
procedure TForm1.SetPenClick(Sender: TObject);
begin
  MapTivate1.Themes[0].Styles[0].Pen.color:=clblue;
  MapTivate1.Themes[0].Styles[0].Pen.width:=1;
  MapTivate1.Themes[0].Styles[0].Pen.style:=3;
  MapTivate1.Themes[0].Styles[0].Pen.BackColor:=clred;
  MapTivate1.Themes[0].Styles[0].Pen.outerwidth:=12;
  MapTivate1.RedrawMap;
end;
```

Width:Integer

Width of the pen in pixels. This is the inner width, when drawing a double-line polyline. Note that only style pxSolid is permitted to have a width greater than one pixel (a Windows API limitation).



Sample Code (Delphi 5)

```
procedure TForm1.ConnectClick(Sender: TObject);
var MyTheme:ITRTheme;
begin
  // Connect to theme and load it when needed to render it
  MyTheme:=MapTivate1.ConnectTheme('mcds.mif',0);
  MessageBeep(0);
  MyTheme.styles[0].Pen.width:=4;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command16_Click()
' Create a Line and a point Using the UserCAD Interface
MapTivate1.UserCad.AddStyle
  MapTivate1.UserCad.StyleNum = MapTivate1.UserCad.StyleCount - 1
  '-------
  With MapTivate1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Color = vbYellow
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Width = 3
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Size = 8
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Upper = 1000
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Visible = True
    .UserCad.CreateLine .MapLeft, .MapTop, .MapRight, .MapBottom
    .UserCad.CreatePoint .MapCenterX - 1 * (.ZoomScale / 120), .MapCenterY - 1 *
(.ZoomScale / 120)
    .RedrawMap
  End With
End Sub
```

9. ITRBrush Object

Setting all the Brush properties.

BackColor:Integer

Brush Background color.

Sample Code (Delphi 5)

```
procedure TForm1.SetBrushClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
  begin
    // Setting Style for fill brush
    Styles[0].Brush.color:=clyellow;
    Styles[0].Brush.BackColor:=clblue;
  end;
  MapTivate1.RedrawMap;
end;
```

BitMap:TRBitMap

Information about the bitmap to be used when a user-defined brush is used. See TRBitMap interface.

Sample Code (Delphi 5)

```
procedure TForm1.SetBrushClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
```

```
  begin
    // Set Brush to user specified bitmap
    Styles[0].Brush.style:=-1;
    with Styles[0].Brush do
    begin
      bitmap.LoadImage('earth3.bmp');
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

Color:Integer

Brush foreround color.

Sample Code (Delphi 5)

```
procedure TForm1.SetBrushClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
  begin
    // Setting Style for fill brush
    Styles[0].Brush.color:=clyellow;
    Styles[0].Brush.BackColor:=clblue;
  end;
```

Sample Code (Visual Basic)

```
Private Sub Command9_Click()
Dim i As Integer
  FoundIndex = -1
  ' Determine which Theme is the World-Countries theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "County-Boundary" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
  ' Change some of the county theme attributes
  ' General RGB Color setting is &HBBGGRR&
  MyCountyBoundary.Styles(0).Brush.Color = vbYellow  ' Set County color
    Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: County Boundary Theme is NOT connected!", vbOKOnly,
"Error was encountered!")
  End If

  FoundIndex = -1
  ' Determine which Theme is the World-Countries theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "County-Water" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
  MyCountyWater.Styles(0).Pen.Color = vbBlue  ' Set Water color
  MyCountyWater.Styles(0).Brush.Color = vbBlue
      Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: County Water Theme is NOT connected!", vbOKOnly,
"Error was encountered!")
  End If

  FoundIndex = -1
  ' Determine which Theme is the World-Countries theme before changing Attributes
  For i = 0 To MapTivate1.ThemeCount - 1
    If MapTivate1.Themes(i).Name = "County-Roads" Then FoundIndex = i
  Next i
  If FoundIndex <> -1 Then
```

```
  ' Set parameters to display Road names at certain scales
    MyCountyRoads.Styles(0).Pen.Color = &H11AABB  ' Set Roads color
    MyCountyRoads.Styles(0).formatstr = "F03"
    MyCountyRoads.Styles(0).Font.Upper = 5
    MyCountyRoads.Styles(0).Font.Lower = 0
    ' Make sure collision detection is ON so labels are readable
    MyCountyRoads.Styles(0).Font.CollisionDetection = True
    ' Label each road in the viewport ONLY once
    MyCountyRoads.Styles(0).Font.UseOnce = True
  Else
    ' Pop Dialog and notify user
    UserResponse = MsgBox("Problem: County Roads Theme is NOT connected!", vbOKOnly,
"Error was encountered!")
  End If
  ' redraw, just in case
  MapTivate1.RedrawMap
End Sub
```

Mode:Integer

Reserved

Style:Integer

An array of brush styles. Some are derived from stock Brush styles. It can take the following values:

Style Value	Style	Delphi Equivalent
-1	User	N/A – Use Bitmap
0		BsSolid
1		BsClear
2		BsBDiagonal
3		BsFDiagonal
4		BsCross
5		BsDiagCross
6		BsHorizontal
7		BsVertical

Sample Code (Delphi 5)

```
procedure TForm1.SetBrushClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
  begin
    // Setting Style for fill brush
    Styles[0].Brush.color:=clyellow;
    Styles[0].Brush.style:=4;
    Styles[0].Brush.BackColor:=clblue;
  end;
```

Transparent:Boolean

Sets the transparent mode of the brush.

10. ITRFont Object

This object is used to set all Font properties for the various themes.

Alignment:Integer

The Text alignment options for the selected font, within the string bounding rectangle, are based on a bit setting, thus allowing the user to combine alignment options. The bit setting correspond to the following alignment options:

Bit Position	Value	Alignment
0	1	Left
1	2	Right
0 & 1	3	Center
2	4	Top
3	8	Bottom
2 & 3	12	Vertical Center

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint1Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Font.height:=-18;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clyellow;
    UserPaint.Font.Alignment:=$F;
    UserPaint.Font.style:=4;
    UserPaint.Rectangle(MapLeft,MapBottom,MapRight,MapTop);
    // Set Aspect Ratio for Vert Ellipse
    AspRatio:=2;
    UserPaint.Circle(MapCenterX+3,MapCenterY+3,3,2);
    UserPaint.Point(MapCenterX-1,MapCenterY+1);
    // Change color and width to draw line
      UserPaint.pen.width:=4;
      UserPaint.pen.color:=clblue;
    UserPaint.Line(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.text(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

Angle:Integer

Rotates the Text that uses this font. Counterclockwise is positive.

Sample Code (Delphi 5)

```
procedure TForm1.FontRoadsClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
  begin
    Styles[0].FormatStr:='+06';
    // Setting Font Attributes
    Styles[0].font.name:='verdana';
    Styles[0].font.color:=clblue;
    Styles[0].font.backcolor:=clyellow;
    Styles[0].font.style:=4;
    Styles[0].font.Upper:=1;
    Styles[0].font.Lower:=0.05;
    Styles[0].font.offsetX:=0;
    Styles[0].font.height:=StrToInt(edit2.text);
    Styles[0].font.alignment:=StrToInt(edit3.text);
    Styles[0].font.style:=StrToInt(edit6.text);
    Styles[0].Font.casing:=strtoint(edit1.text);
    Styles[0].Font.Angle:=strtoint(edit5.text);
  end;
  MapTivate1.RedrawMap;
end;
```

BackColor:Integer

The background Font Color. Only effective when shadow text option is selected by the user.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint1Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Font.height:=-18;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clyellow;
    UserPaint.Font.Alignment:=strtoint(edit4.text);
    UserPaint.Font.style:=4;
    // Change color and width to draw line
      UserPaint.pen.width:=4;
      UserPaint.pen.color:=clblue;
    UserPaint.Line(MapLeft,MapBottom,MapRight,MapTop);
    UserPaint.text(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

Casing:Integer

Defines the casing to be used for the text using this font. The options are:

- No Effect
- All Lower Case
- All Upper Case

- Proper Case

Sample Code (Visual Basic)

```
Private Sub Command42_Click()
  nFontCase = nFontCase + 1
  If nFontCase > 3 Then nFontCase = 0
  For i = 0 To MapTivate1.Themes(0).StyleCount / 2
    ' Change the casing
    MapTivate1.Themes(0).Styles(i).Font.Casing = nFontCase
    ' Also change the color and size so they stand out
    MapTivate1.Themes(0).Styles(i).Font.Color = vbGreen
    MapTivate1.Themes(0).Styles(i).Font.Height = -20
  Next i
  MapTivate1.RedrawMap
End Sub
```

CollisionDetection:Boolean

Defines whether to use collision detection when labeling objects assigned to the current style. Collision detection is based on the bounding polygons of the labels. If set to false, text is immediately placed at its reference point and is permitted to overlap other labels.

Color:Integer

The foreground font color.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint1Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Font.height:=-18;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clyellow;
    UserPaint.Font.Alignment:=strtoint(edit4.text);
    UserPaint.Font.style:=4;
    UserPaint.text(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

Frame:Boolean

Defines whether a black frame will be drawn outlining the bounding polygon for each drawn label.

Sample Code (Visual Basic)

```
Private Sub Command42_Click()
  nFontCase = nFontCase + 1
  If nFontCase > 3 Then nFontCase = 0
  For i = 0 To MapTivate1.Themes(0).StyleCount / 2
    ' Change the casing
    MapTivate1.Themes(0).Styles(i).Font.Casing = nFontCase
    ' Also change the color and size so they stand out
    MapTivate1.Themes(0).Styles(i).Font.Color = vbBlue
    MapTivate1.Themes(0).Styles(i).Font.Height = -18
```

```
    ' Draw frame to denote bounding box for collision detection
    MapTivate1.Themes(0).Styles(i).Font.Frame = True
  Next i
  MapTivate1.RedrawMap
End Sub
```

Height:Integer

If the value is positive, it represents the height of the font in pixels. If the value is negative, the font point size.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint1Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Font.height:=-22;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clyellow;
    UserPaint.Font.Alignment:=strtoint(edit4.text);
    UserPaint.Font.style:=4;
    UserPaint.text(MapCenterX+1,MapCenterY-1,'Sample Text 2 ');
    RefreshMap;
  end;
end;
```

Lower:Double

The lower scale at which the text using this font is visible. Also see ***Upper*** property.

Sample Code (Delphi 5)

```
procedure TForm1.NewFontClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
  begin
    // Setting Font Attributes
    Styles[0].font.name:='verdana';
    Styles[0].font.color:=clblue;
    Styles[0].font.backcolor:=clyellow;
    Styles[0].font.style:=4;
    Styles[0].font.Upper:=0.5;
    Styles[0].font.Lower:=0.05;
    Styles[0].font.height:=StrToInt(edit2.text);
    Styles[0].font.alignment:=StrToInt(edit3.text);
  end;
  MapTivate1.RedrawMap;
end;
```

Name:String

The name of the font (Arial, Verdana, Times Roman, etc.)

Sample Code (Delphi 5)

```
procedure TForm1.FontRoadsClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
```

```
  begin
    Styles[0].FormatStr:='+06';
    // Setting Font Attributes
    Styles[0].font.name:='verdana';
    Styles[0].font.color:=clblue;
    Styles[0].font.backcolor:=clyellow;
    Styles[0].font.style:=4;
    Styles[0].font.Upper:=1;
    Styles[0].font.Lower:=0.05;
    Styles[0].font.offsetX:=0;
    Styles[0].font.height:=StrToInt(edit2.text);
    Styles[0].font.alignment:=StrToInt(edit3.text);
    Styles[0].font.style:=StrToInt(edit6.text);
  end;
  MapTivate1.RedrawMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command42_Click()
  nFontCase = nFontCase + 1
  If nFontCase > 3 Then nFontCase = 0
  For i = 0 To MapTivate1.Themes(0).StyleCount / 2
    ' Change the casing
    MapTivate1.Themes(0).Styles(i).Font.Casing = nFontCase
    ' Also change the color and size so they stand out
    MapTivate1.Themes(0).Styles(i).Font.Color = vbBlue
    MapTivate1.Themes(0).Styles(i).Font.Height = -18
    ' Draw frame to denote bounding box for collision detection
    MapTivate1.Themes(0).Styles(i).Font.Frame = True
    ' Finally, change the font name.  Note that specifying "Comic Sans" finds no match!
    MapTivate1.Themes(0).Styles(i).Font.Name = "Comic Sans MS"
  Next i
  MapTivate1.RedrawMap
End Sub
```

OffsetX:Integer

The X-Offset, in pixels, for placement of user-strings in text bounding polygons. The specified number of horizontal pixels is added to the current reference point of the string being displayed.

OffsetY:Integer

The Y-Offset, in pixels, for placement of user-strings in text bounding polygons. The specified number of vertical pixels is added to the current reference point of the string being displayed.

Style:Integer

The font styles avainable are set by a bit position, i.e., they can be combined. They are as follows:

Value	Effect	
1	Bold	← Any of these effects May be combined
2	Italic	
4	Underline	
8	Strikethrough	
16	Shadow	←These effects are

32	Lowered	mutually exclussive
48	Raised	

Effects are additive, for example 3 would be Italic-Bold, 17 bold-shadow, etc.). ***Note*** that the shadow setting has no bearing when labeling polylines, it only applies to labeling points.

Sample Code (Delphi 5)

```
procedure TForm1.NewFontClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
  begin
    // Setting Font Attributes
    Styles[0].font.name:='verdana';
    Styles[0].font.color:=clblue;
    Styles[0].font.backcolor:=clyellow;
    Styles[0].font.style:=4;
    Styles[0].font.Upper:=0.5;
    Styles[0].font.Lower:=0.05;
    Styles[0].font.height:=StrToInt(edit2.text);
    Styles[0].font.alignment:=StrToInt(edit3.text);
  end;
  MapTivate1.RedrawMap;
end;
```

Upper:double

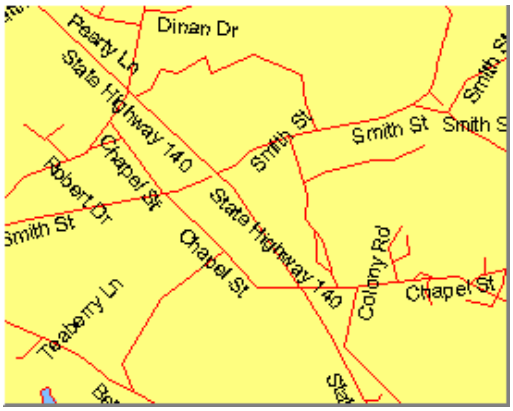
Upper scale at which the text using this font is visible. This controls the labeling threshold. For example, if you had a theme MyRoads and had set MyRoads.Upper = 10 and MyRoad.Font.Upper = 2, then the roads would become visible when the scale was < 10, but the roads would not be labeled until the scale became < 2.

Sample Code (Delphi 5)

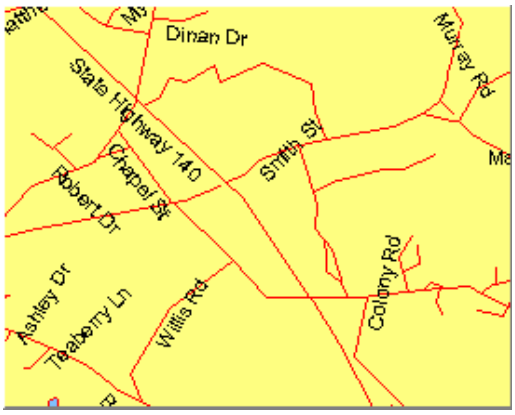
```
procedure TForm1.NewFontClick(Sender: TObject);
begin
  with MapTivate1.Themes[0] do
  begin
    // Setting Font Attributes
    Styles[0].font.name:='verdana';
    Styles[0].font.Upper:=0.5;
    Styles[0].font.Lower:=0.05;
    Styles[0].font.height:=StrToInt(edit2.text);
    Styles[0].font.alignment:=StrToInt(edit3.text);
  end;
  MapTivate1.RedrawMap;
end;
```

UseOnce:Boolean

If set to true, then a check is made to determine if the name about to be painted has already been used to label a segment in the viewport, and if so, it is rejected. Otherwise, multiple segments of the same object may be labeled in the viewport.



UseOnce=False



UseOnce=True

Visible:Boolean

Determines the visibility of the text using this font.

11. ITRMark Object

Controls the attributes of the mark that is used to identify mapping elements.

BorderColor:Integer

The color of the border of the mark object being used.

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
with MapTivate1.Themes[0] do
  begin
    // Setting Mark Attributes
    // Mark Polygon Centroids
    Styles[0].Mark.VisFilter:=4;
    Styles[0].Mark.Visible:=true;
    Styles[0].Mark.Style:=2;
    Styles[0].Mark.Upper:=20;
    Styles[0].Mark.Lower:=0.1;
    Styles[0].Mark.Color:=clBlue;
    Styles[0].Mark.BorderColor:=clRed;
    Styles[0].Mark.Size:=14;
  end;
  MapTivate1.RedrawMap;
end;
```

BitMap:TRBitMap

Sets the properties for the bitmap to be used for Marks, ir Mark.Style is set to –1. See TRBitMap interface.

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
with MapTivate1.Themes[0] do
  begin
    // Setting Mark Attributes
    // Mark Polygon Centroids
    with styles[0].mark do
    begin
      VisFilter:=4;
      Visible:=true;
      Style:=-1;
      bitmap.LoadImage('redbtn.bmp');
      Upper:=20;
      Lower:=0.1;
      Color:=clBlue;
      BorderColor:=clRed;
      Size:=14;
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

Color:Integer

The fill color of the mark object being used. Applies to Mark.Styles > -1.

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
inc(IMrkStyle);
If IMrkStyle>6 then IMrkStyle:=0;
with MapTivate1.Themes[0] do
  begin
    styles[0].brush.style:=1;
    // Setting Mark Attributes
    with styles[0].mark do
    begin
      VisFilter:=4;
      Visible:=true;
      Style:=IMrkStyle;
      Upper:=20;
      Lower:=0.1;
      Color:=clBlue+IMrkStyle*100;
      BorderColor:=clRed;
      Size:=10+IMrkStyle*4;
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

SymbolFont:String

The name of the TT font to be used for selecting the marker symbol, see *.symbol*, below. Only applicable if Mark.Style is set to –2.

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
with MapTivate1.Themes[0] do
  begin
    styles[0].brush.style:=1;
    // Setting Mark Attributes to mark Polygon Centroids
    with styles[0].mark do
    begin
      VisFilter:=7;
      Visible:=true;
      Style:=-2;
      SymbolFont:='Symbol';
      Symbol:=80;
      Upper:=20;
      Lower:=0.1;
      Color:=clBlue;
      BorderColor:=clRed;
      Size:=14;
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

Lower:double

The lower scale at which the mark will be visible.

Size:Integer

The size of the mark's bounding box in pixels. Note that: width=height=size.

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
inc(IMrkStyle);
If IMrkStyle>6 then IMrkStyle:=0;
with MapTivate1.Themes[0] do
  begin
    styles[0].brush.style:=1;
    // Setting Mark Attributes
    with styles[0].mark do
    begin
      VisFilter:=4;
      Visible:=true;
      Style:=IMrkStyle;
      Upper:=20;
      Lower:=0.1;
      Color:=clBlue+IMrkStyle*100;
      BorderColor:=clRed;
      Size:=10+IMrkStyle*4;
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

Style:Integer

Identifies the mark style number to use. The following styles are available:

-2	Use Symbol from TT font
-1	Use user-defined Bitmap
0	Rectangle
1	Diamond
2	Ellipse/Circle
3	Up Pointing triangle
4	Down pointing triangle
5	Iframe character
6	Cross

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
with MapTivate1.Themes[0] do
  begin
    styles[0].brush.style:=1;
    // Setting Mark Attributes
    with styles[0].mark do
    begin
      VisFilter:=7;
      Visible:=true;
      Style:=3;
      Upper:=20;
      Lower:=0.1;
      Color:=clBlue;
      BorderColor:=clRed;
      Size:=14;
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

Symbol:LongInt

The character code, from the specified TT font, to be used as a mark, when Mark.Style is set to –2.

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
with MapTivate1.Themes[0] do
  begin
    styles[0].brush.style:=1;
    // Setting Mark Attributes
    with styles[0].mark do
    begin
      Style:=-2;
      VisFilter:=4;
      Visible:=true;
      SymbolFont:='Symbol';
      Symbol:=80;
      Upper:=20;
      Lower:=0.1;
      Color:=clBlue;
```

```
      BorderColor:=clRed;
      Size:=14;
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

Upper:Double

The upper scale at which the mark object will be visible.

Visible:Boolean

Sets the visibility of the mark object.

VisFilter:Integer

Determines what objects the specified Mark will be associated with (what objects it will mark). Based on bit position.

Bit position 1 set (value = 1) - Mark Points (Marker placed at the point coordinates).
Bit position 2 set (value = 2) - Mark Polylines (Marker placed at the polyline center point).
Bit position 3 set (value = 4) - Mark Polygons Mark Polylines (Marker placed at the polygon centroid).

If VisFilter is Set to 7 (all 3 bits set) then all three types of objects are marked with the specified mark.

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
with MapTivate1.Themes[0] do
  begin
    styles[0].brush.style:=1;
    // Setting Mark Attributes
    with styles[0].mark do
    begin
      VisFilter:=7;
      Visible:=true;
      Style:=3;
      Upper:=20;
      Lower:=0.1;
      Color:=clBlue;
      BorderColor:=clRed;
      Size:=14;
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

12. ITRCAD Object

Enables the user to create persistent user-defined primitives, that are drawn on top of the map surface of MapTivate, as a special purpose Theme. Note that these objects are different than those created using the UserPaint Objects in that they are part of the map and get redrawn every time the map is drawn, whereas with UserPaint, the user is responsible for refreshing the map surface on their own.

AddStyle():Integer;

Adds a style to the list of defined styles (appends it to the end of the list) and increments StyleCount. Note that a Style needs to be added before its attributes can be modified. The only exception is Style[0], which is the default style and is pre-defined.

Clear()

Clears all objects in the current CAD Theme and frees up all resources.

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.USerCad.Clear;
  MapTivate1.UserCad.LoadFromFile('MyCad01');
  n:=MapTivate1.usercad.Count;
  MapTivate1.RefreshMap;
end;
```

ClearStyles()

Clears all currently defined styles in the Theme.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  // Delete Style # 5
  MapTivate1.UserCAD.DeleteStyle(5);
  // Clear Current Styles
  MapTivate1.UserCAD.ClearStyles;
end;
```

Count:Integer

The number of objects in the current UserCAD theme.

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.USerCad.Clear;
  MapTivate1.UserCad.LoadFromFile('MyCad01');
  n:=MapTivate1.usercad.Count;
  MapTivate1.RefreshMap;
end;
```

CreateEllipse(X,Y:Double; R:Double; AspRatio:Double);

Creates an ellipse based on the parameters specified by the user.

X,Y	Centerpoint coordinates (Lon/Lat)
R	Horizontal Radius (degrees)
AspRatio	Aspect Ratio (Vertical to Horizontal radius)

Sample Code (Delphi 5)

```
procedure TForm1.CADEllipseClick(Sender: TObject);
var x,y,radius,ARatio:double;
begin
  x:=-70;
  y:=42;
  Radius:=1;
  ARatio:=2;
  // Create Ellipse
  MapTivate1.UserCad.CreateEllipse(x,y,Radius,ARatio);
  MapTivate1.RedrawMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command12_Click()
' Draw an ellipse using the UserCAD interface
    ' StyleNum needs to be set before attributes are assigned
    MapTivate1.UserCad.StyleNum = 0
    MapTivate1.UserCad.Styles(0).Brush.Color = vbBlue
    MapTivate1.UserCad.Styles(0).Pen.Color = clblue
    MapTivate1.UserCad.Styles(0).Brush.Style = 7
    MapTivate1.UserCad.Styles(0).Brush.Transparent = True
    MapTivate1.UserCad.CreateEllipse MapTivate1.MapCenterX, MapTivate1.MapCenterY, 1.25 *
(MapTivate1.ZoomScale / 40), 0.2
    MapTivate1.RedrawMap
End Sub
```

CreateLine(X1,Y1,X2,Y2: Double)

Creates a line between the two points specified by the user.

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.UserCad.CreatePoint(-80,42);
  MapTivate1.UserCad.CreateLine(-80,43,-79,43);
  MapTivate1.UserCad.SaveToFile('MyCad02');
  MapTivate1.RedrawMap;
```

```
end;
```

Sample Code (Visual Basic)

```
Private Sub Command16_Click()
' Create a Line and a point Using the UserCAD Interface
MapTivate1.UserCad.AddStyle
  MapTivate1.UserCad.StyleNum = MapTivate1.UserCad.StyleCount - 1
  '-------
  With MapTivate1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Color = vbYellow
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Width = 3
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Size = 8
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Upper = 1000
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Visible = True
    .UserCad.CreateLine .MapLeft, .MapTop, .MapRight, .MapBottom
    .UserCad.CreatePoint .MapCenterX - 1 * (.ZoomScale / 120), .MapCenterY - 1 *
(.ZoomScale / 120)
    .RedrawMap
  End With
  '-------
End Sub
```

CreatePoint(X,Y: double)

Create a point object at the coordinates specified by the user.

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.UserCad.CreatePoint(-80,42);
  MapTivate1.UserCad.CreatePoint(-81,42);
  MapTivate1.UserCad.CreateText(-80.5,41,'Test');
  MapTivate1.UserCad.SaveToFile('MyCad01');
  MapTivate1.USerCad.Clear;
  MapTivate1.RefreshMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command16_Click()
' Create a Line and a point Using the UserCAD Interface
MapTivate1.UserCad.AddStyle
  MapTivate1.UserCad.StyleNum = MapTivate1.UserCad.StyleCount - 1
  '-------
  With MapTivate1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Color = vbYellow
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Width = 3
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Size = 8
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Upper = 1000
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Mark.Visible = True
    .UserCad.CreateLine .MapLeft, .MapTop, .MapRight, .MapBottom
    .UserCad.CreatePoint .MapCenterX - 1 * (.ZoomScale / 120), .MapCenterY - 1 *
(.ZoomScale / 120)
    .RedrawMap
  End With
  '-------
End Sub
```

CreatePolygon(var Points:PtTRPoint; Var Counts:Integer; Num:Integer)

Creates a polygon object (may be composed of multiple polygons).

Points TRPoint Array
Counts Counts Array
Num Number of polygons in the object

Sample Code (Delphi 5)

```
 procedure TForm1.CADPolyGonClick(Sender: TObject);
var points:array of trpoint;
    counts:array of integer;
    n:integer;
begin
  setlength(points,15);
  setlength(counts,3);
  points[0].x:=-70;  points[0].y:=44;
  points[1].x:=-72;  points[1].y:=42;
  points[2].x:=-74;  points[2].y:=44;
  points[3].x:=-74;  points[3].y:=40;
  points[4].x:=-72;  points[4].y:=44;
  points[5].x:=-70;  points[5].y:=40;
  points[6].x:=-68;  points[6].y:=46;
  points[7].x:=-68;  points[7].y:=38;
  points[8].x:=-76;  points[8].y:=38;
  points[9].x:=-76;  points[9].y:=46;
  points[10].x:=-68;  points[10].y:=46;
  counts[0]:=3;
  counts[1]:=3;
  counts[2]:=5;
  // Note that based on the way Windows treats overlapping bruches...
  // First poly in object is painted with solid red
  // Second poly is painted in solid red, areas overlaping the
  // first polygon are clear (hollow).
  // Third poly is painted solid read.  Areas overlapping either
  // poly #1 or poly #2 are clear.  Areas overlapping sections common
  // to poly #1 and poly #2 (which were clear above), are painted solid...
   MapTivate1.UserCad.StyleNum:=0;
   MapTivate1.UserCad.CreatePolygon(points[0] , counts[0], 3);
  n:=MapTivate1.usercad.Count;
  MapTivate1.UserCad.SaveToFile('MyCad01');
  MapTivate1.USerCad.Clear;
  n:=MapTivate1.usercad.Count;
  MapTivate1.UserCad.LoadFromFile('MyCad01');
  MapTivate1.RedrawMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command13_Click()
' Create a Polygon using the UserCAD Interface
Dim i As Integer, count As Long
Dim MCnt(0 To 4) As Long
Dim MPts(0 To 7) As TRPoint
  ' Create the points to be used for the polygon
  ' Set them relative to the currect ZoomScale value
  With MapTivate1
    MPts(0).X = .MapCenterX - 1.1 * (.ZoomScale / 120)
    MPts(0).Y = .MapCenterY - 1.1 * (.ZoomScale / 120)
    MPts(1).X = .MapCenterX - 3.1 * (.ZoomScale / 120)
    MPts(1).Y = .MapCenterY + 2.1 * (.ZoomScale / 120)
    MPts(2).X = .MapCenterX - 3.05 * (.ZoomScale / 120)
    MPts(2).Y = .MapCenterY + 2.5 * (.ZoomScale / 120)
```

```
    MPts(3).X = .MapCenterX - 0.5 * (.ZoomScale / 120)
    MPts(3).Y = .MapCenterY + 1.7 * (.ZoomScale / 120)
    MPts(4).X = .MapCenterX + 1.1 * (.ZoomScale / 120)
    MPts(4).Y = .MapCenterY - 1.1 * (.ZoomScale / 120)
  End With
  MCnt(0) = 5
  count = 1
  ' Add a style
  MapTivate1.UserCad.AddStyle
  ' Set Attributes for the newly added style
  MapTivate1.UserCad.StyleNum = MapTivate1.UserCad.StyleCount - 1
  With MapTivate1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Brush.Color = vbBlue
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Brush.BackColor = vbYellow
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Color = vbBlue
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Brush.Style = 3
    ' Create the polygon
    .UserCad.CreatePolygon MPts(0), MCnt(0), count
    .RedrawMap
  End With
    ' Zoom to the extents of the USerCAD Theme
    MapTivate1.ZoomMapRect MapTivate1.UserCad.Extents.Xmin,
MapTivate1.UserCad.Extents.Ymin, MapTivate1.UserCad.Extents.Xmax,
MapTivate1.UserCad.Extents.Ymax
End Sub
```

CreatePolyLine(var Points:PtTRPoint; Var Counts:Integer; Num:Integer)

Creates a polyline of multiple segments.

Points	TRPoint Array
Counts	Counts Array
Num	Number of segments in the object

Sample Code (Delphi 5)

```
procedure TForm1.CADPolyLineClick(Sender: TObject);
var points:array of trpoint;
    counts:array of integer;
    n:integer;
begin
  setlength(points,15);
  setlength(counts,3);
  points[0].x:=-70;  points[0].y:=44;
  points[1].x:=-72;  points[1].y:=42;
  points[2].x:=-74;  points[2].y:=44;
  points[3].x:=-74;  points[3].y:=40;
  points[4].x:=-72;  points[4].y:=44;
  points[5].x:=-70;  points[5].y:=40;
  points[6].x:=-68;  points[6].y:=46;
  points[7].x:=-68;  points[7].y:=38;
  points[8].x:=-76;  points[8].y:=38;
  points[9].x:=-76;  points[9].y:=46;
  points[10].x:=-68;  points[10].y:=46;
// Thre segment polyline each segment with 3, 3, and 5 points respectively
  counts[0]:=3;
  counts[1]:=3;
  counts[2]:=5;
   MapTivate1.UserCAD.styles[0].pen.color:=clblue;
   MapTivate1.UserCAD.styles[0].Pen.width:=2;
   MapTivate1.UserCAD.styles[0].pen.backcolor:=clyellow;
   MapTivate1.UserCAD.styles[0].Pen.outerwidth:=4;
   MapTivate1.UserCAD.styles[0].brush.color:=clred;
   MapTivate1.UserCAD.styles[0].brush.style:=integer(bssolid);
   MapTivate1.UserCad.StyleNum:=0;
```

```
   MapTivate1.UserCad.CreatePolyline(points[0] , counts[0], 3);
  MapTivate1.RedrawMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command17_Click()
' Create a Polyline using the UserCAD Interface
Dim i As Integer, count As Long
Dim MCnt(0 To 4) As Long
Dim MPts(0 To 7) As TRPoint
  '-------
  With MapTivate1
    MPts(0).X = .MapCenterX - 1.1 * (.ZoomScale / 120)
    MPts(0).Y = .MapCenterY - 1.1 * (.ZoomScale / 120)
    MPts(1).X = .MapCenterX - 3.1 * (.ZoomScale / 120)
    MPts(1).Y = .MapCenterY + 2.1 * (.ZoomScale / 120)
    MPts(2).X = .MapCenterX - 3.15 * (.ZoomScale / 120)
    MPts(2).Y = .MapCenterY + 2.6 * (.ZoomScale / 120)
    MPts(3).X = .MapCenterX - 1.5 * (.ZoomScale / 120)
    MPts(3).Y = .MapCenterY + 1.1 * (.ZoomScale / 120)
    MPts(4).X = .MapCenterX + 1.8 * (.ZoomScale / 120)
    MPts(4).Y = .MapCenterY + 4.2 * (.ZoomScale / 120)
  End With
  '-------
  MCnt(0) = 5
  count = 1
  ' Add a style before cmodifying it's attributes
  MapTivate1.UserCad.AddStyle
  MapTivate1.UserCad.StyleNum = MapTivate1.UserCad.StyleCount - 1
  With MapTivate1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Color = vbBlue
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Width = 1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.BackColor = vbYellow
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.OuterWidth = 5
    .UserCad.CreatePolyline MPts(0), MCnt(0), count
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Font.Style = 2
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Font.Color = vbRed
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Font.Height = -18
    .UserCad.CreateText .MapCenterX, .MapCenterY, "CAD Text Object"
    .RedrawMap
  End With
End Sub
```

CreateRectangle(X1,Y1,X2,Y2: double)

Creates a rectangle between the upper left and lower right corner points specified by the user.

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.UserCad.CreateText(-80,40,'Testing the CAD Text 01');
  MapTivate1.UserCad.CreateRectangle(-80,44,-79,45);
  MapTivate1.UserCad.CreateRegPolygon(-80,43,-79,44,3);
  MapTivate1.RedrawMap;
end;
```

CreateRegPolygon(X1,Y1,X2,Y2:double; N:Integer)

Creates a regular polygon inscribed in the circle defined by the centerpoint X1,Y1 and a point on the circumference through X2,Y2. The first vertex is at point X2,Y2.

X1,Y1 - Centerpoint coordinates
X2,Y2 - Coordinates of first point on circumference
N - Number of vertices

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.UserCad.CreateText(-80,40,'Testing the CAD Text 01');
  MapTivate1.UserCad.CreateRectangle(-80,44,-79,45);
  MapTivate1.UserCad.CreateRegPolygon(-80,43,-79,44,3);
  MapTivate1.RedrawMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command15_Click()
' Create a regular Polygon in UserCAD
Dim j As Integer
  MapTivate1.UserCad.AddStyle
  MapTivate1.UserCad.StyleNum = MapTivate1.UserCad.StyleCount - 1
  '-------
  With MapTivate1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Brush.Color = vbGreen
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Brush.BackColor = vbgray
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Color = vbBlue
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Brush.Style = 3
    .UserCad.CreateRegPolygon .MapCenterX - 1 * (.ZoomScale / 120), .MapCenterY - 1 *
(.ZoomScale / 120), .MapCenterX - 3 * (.ZoomScale / 120), .MapCenterY - 3 * (.ZoomScale /
120), 5
  End With
  '-------
  MapTivate1.RedrawMap
End Sub
```

CreateText(x,y:double; s:widestring)

Create a text object using the coordinates and the string specified by the user.

X,Y - Coordinates for placing the string
S - String to be placed

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.UserCad.CreatePoint(-80,42);
  MapTivate1.UserCad.CreateText(-80,40,'Testing the CAD Text 01');
  MapTivate1.UserCad.CreateText(-80,41,'Testing the CAD Text 02');
  MapTivate1.UserCad.CreateText(-80,42,'Testing the CAD Text 03');
  MapTivate1.RefreshMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command17_Click()
' Create a Polyline using the UserCAD Interface
Dim i As Integer, count As Long
Dim MCnt(0 To 4) As Long
Dim MPts(0 To 7) As TRPoint
  '-------
  With MapTivate1
    MPts(0).X = .MapCenterX - 1.1 * (.ZoomScale / 120)
    MPts(0).Y = .MapCenterY - 1.1 * (.ZoomScale / 120)
    MPts(1).X = .MapCenterX - 3.1 * (.ZoomScale / 120)
    MPts(1).Y = .MapCenterY + 2.1 * (.ZoomScale / 120)
    MPts(2).X = .MapCenterX - 3.15 * (.ZoomScale / 120)
    MPts(2).Y = .MapCenterY + 2.6 * (.ZoomScale / 120)
    MPts(3).X = .MapCenterX - 1.5 * (.ZoomScale / 120)
    MPts(3).Y = .MapCenterY + 1.1 * (.ZoomScale / 120)
    MPts(4).X = .MapCenterX + 1.8 * (.ZoomScale / 120)
    MPts(4).Y = .MapCenterY + 4.2 * (.ZoomScale / 120)
  End With
  '-------
  MCnt(0) = 5
  count = 1
  ' Add a style before cmodifying it's attributes
  MapTivate1.UserCad.AddStyle
  MapTivate1.UserCad.StyleNum = MapTivate1.UserCad.StyleCount - 1
  With MapTivate1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Color = vbBlue
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.Width = 1
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.BackColor = vbYellow
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Pen.OuterWidth = 5
    .UserCad.CreatePolyline MPts(0), MCnt(0), count
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Font.Style = 2
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Font.Color = vbRed
    .UserCad.Styles(MapTivate1.UserCad.StyleNum).Font.Height = -18
    .UserCad.CreateText .MapCenterX, .MapCenterY, "CAD Text Object"
    .RedrawMap
  End With
End Sub
```

DeleteObject(Index, Num:Integer);

Delete Num obects starting with Index.

Sample Code (Delphi 5)

```
procedure TForm1.DelCAD2Click(Sender: TObject);
begin
  // Delete the third object (zero based)
  MapTivate1.UserCad.DeleteObject(2,1);
end;
```

DeleteStyle(N:Integer)

Deletes the specified Style from the UserCAD object.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  // Delete Style # 5
  MapTivate1.UserCAD.DeleteStyle(5);
  // Clear Current Styles
  MapTivate1.UserCAD.ClearStyles;
end;
```

Enabled:Boolean

Defines whether the UserCAD theme is enabled (True) or disabled (False).

Extents:TRExtents

A record containing the extents of the current USerCAD theme. (Read-Only)

Sample Code (Delphi 5)

```
procedure TForm1.UserCADExtentsClick(Sender: TObject);
var CadExt:TRExtents;
begin
  CadExt:=MapTivate1.UserCAD.Extents;
  // Display current Extents
  Label2.caption:=floattostr(CadExt.xmin);
  Label3.caption:=floattostr(CadExt.ymin);
  Label4.caption:=floattostr(CadExt.xmax);
  Label5.caption:=floattostr(CadExt.ymax);
  MessageBeep(0);
  MapTivate1.ZoomMapRect(CadExt.xmin,CadExt.ymin,CadExt.xmax,CadExt.ymax);
end;
```

InsertStyle(N:LongInt);

Inserts a style immediately following the style ‘N’ specified by the user.

Items[Index:Integer]:TRCADObject

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var kk,n:integer;
ck:String
begin
    for kk:=0 to MapTivate1.UserCad.Count-1 do
    begin
      ck:=Inttostr(kk)+' - Not';
      if (MapTivate1.UserCad.Items[kk].selected=true) then
             ck:=Inttostr(kk)+' - Selected';
      ListBox1.items.add(ck);
    end;
end;
```

LoadFromFile(FileName:Widestring);

Load a CAD Theme from the user specified file. The objects in the CAD theme file are added to those already in memory. If that is not desired, then the Clear method may be called prior to loading the data from the file. The default filename extension is .TRC.

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.UserCad.CreatePoint(-80,42);
  MapTivate1.UserCad.LoadFromFile('MyCad01');
  MapTivate1.RefreshMap;
end;
```

Lower:Double

Lower limit of visibility for the UserCAD Theme.

SaveToFile

Save the current CAD Theme definition to a user-specified file. The file is flat ASCII and may be edited by the user using any text editor. See Appendix "C" for a sample listing of a UserCAD file.

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.UserCad.CreatePoint(-80,42);
  MapTivate1.UserCad.SaveToFile('MyCad01');
  MapTivate1.ReDrawMap;
end;
```

SelectAll(Option:Integer)

Perform the specified Select Operation on ALL the objects in the UserCAD Theme. Option=0 – Deselect Objects, Option=1 – Select Objects, Object=2 – Toggle current Selection state.

Sample Code (Delphi 5)

```
procedure TForm1.SelCADAllClick(Sender: TObject);
begin
  // Select all Objects
  MapTivate1.UserCAD.SelectAll(2);
  // Delete all Selected objects
  MapTivate1.UserCAD.SelectDelete;
end;
```

SelectDelete()

Delete all currently selected objects.

Sample Code (Delphi 5)

```
procedure TForm1.SelCADAllClick(Sender: TObject);
begin
  // Select all Object in the specified Rectangle
  MapTivate1.UserCAD.SelectRectangle(-70,32,-100,52,2);
  // Delete all Selected objects
  MapTivate1.UserCAD.SelectDelete;
end;
```

SelectedStyle:integer

Identifies the Style that is used to attribute UserCAD selected items..

SelectPoint(X,Y:Double, Option:Integer)

Select all USerCAD objects going through the point specified by the user.

X,Y Point coordinates (lon/Lat)
Option 0-Deselect, 1-Select, 2-Toggle current selection status

Sample Code (Delphi 5)

```
procedure TForm1.CADSelPtClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.Tolerance:=16;
  MapTivate1.UserCad.SelectPoint(-79,44,2);
  messagebeep(0);
end;
```

SelectRange(Index,Num:Integer; Option:Integer);

Select a range of objects in the current UserCAD theme.

Index Starting Object Number (zero based)
Num Number of objects to select
Option 0-Deselect, 1-Select, 2-Toggle current selection status

Sample Code (Delphi 5)

```
procedure TForm1.SelMvBackClick(Sender: TObject);
begin
  // Select Objects
  MapTivate1.UserCad.SelectRange(2,2,1);
  // Move selected objects to back
  MapTivate1.UserCad.SelectToBack;
end;
```

SelectRectangle(X1,y1,x2,y2: Double; Option:Integer);

Perform the indicated select operation on all objects inside the specified rectangle.

Sample Code (Delphi 5)

```
procedure TForm1.SelCADAllClick(Sender: TObject);
begin
  // Select all Object in the specified Rectangle
  MapTivate1.UserCAD.SelectRectangle(-70,32,-100,52,2);
  // Delete all Selected objects
  MapTivate1.UserCAD.SelectDelete;
end;
```

SelectToBack()

Moves all selected objects behind other UserCAD objects in the theme.

Sample Code (Delphi 5)

```
procedure TForm1.SelMvBackClick(Sender: TObject);
begin
  // Select Objects
  MapTivate1.UserCad.SelectRange(2,2,1);
  // Move selected objects to back
  MapTivate1.UserCad.SelectToBack;
end;
```

SelectToFront()

Moves all selected objects in front of other UserCAD objects in the theme.

Sample Code (Delphi 5)

```
procedure TForm1.SelMvBackClick(Sender: TObject);
begin
  // Select Objects
  MapTivate1.UserCad.SelectRange(0,2,1);
  // Move selected objects to front
  MapTivate1.UserCad.SelectToFront;
end;
```

SelectToStyle(N:Integer)

Moves selected UserCAD objects to the specified Style.

Sample Code (Delphi 5)

```
procedure TForm1.CADMoveClick(Sender: TObject);
begin
  // Move selected objects to style 2
   MapTivate1.UserCAD.selectToSyle(2);
end;
```

StyleCount:Integer

The total number of currently defined styles in the Theme.

StyleNum:Integer

Sets the current Style number. Any CAD objects created subsequent to this will have the attributes defines by StyleNum, until it is set to some other value.

Sample Code (Delphi 5)

```
procedure TForm1.CADObjClick(Sender: TObject);
var n:integer;
begin
  MapTivate1.UserCad.Upper:=3000;
  MapTivate1.UserCad.lower:=0.1;
  MapTivate1.USerCAD.Styles[0].font.color:=clblue;
  MapTivate1.USerCAD.Styles[0].font.style:=3;
  MapTivate1.UserCad.CreateText(-73.0,40.5,'CAD Line');
  // Add Styles before modifying them
  MapTivate1.UserCAD.Addstyle;
  MapTivate1.UserCAD.Addstyle;
  MapTivate1.UserCAD.StyleNum:=2;
  MapTivate1.USerCAD.Styles[2].font.height:=-13;
  MapTivate1.USerCAD.Styles[2].font.color:=clgreen;
  MapTivate1.USerCAD.Styles[2].font.style:=19;
  MapTivate1.UserCad.CreateText(-71.6,42.2,'Labeling with CAD Object');
  MapTivate1.UserCad.styles[2].Brush.style:=5;
  MapTivate1.UserCad.styles[2].Brush.color:=0;
  MapTivate1.UserCad.styles[2].Brush.backcolor:=clnone;
  MapTivate1.UserCad.CreateEllipse(-71.6,42.0,0.5,1);
  MapTivate1.UserCAD.StyleNum:=0;
  MapTivate1.UserCad.CreateText(-80,42,'Testing the CAD Text 03');
  // Define Two Styles
  MapTivate1.USerCAD.Styles[0].pen.color:=clblue;
  MapTivate1.USerCAD.Styles[0].pen.width:=3;
  MapTivate1.USerCAD.Styles[0].Font.height:=12;
  MapTivate1.USerCAD.Styles[1].pen.color:=clyellow;
  MapTivate1.USerCAD.Styles[1].pen.width:=12;
  MapTivate1.USerCAD.Styles[1].Font.height:=28;
  MapTivate1.UserCAD.StyleNum:=0;
  MapTivate1.UserCad.CreateLine(-71.6,42.4,-71.3,42.2);
  MapTivate1.UserCAD.StyleNum:=1;
  MapTivate1.UserCad.CreateLine(-80.5,46,-70,41.5);
  // Change pen, brush and draw rectangle
  MapTivate1.USerCAD.Styles[2].Pen.color:=clgreen;
  MapTivate1.USerCAD.Styles[2].Pen.width:=4;
  MapTivate1.USerCAD.Styles[2].brush.style:=5;
  MapTivate1.USerCAD.Styles[2].brush.color:=clblue;
  MapTivate1.UserCAD.StyleNum:=2;
  MapTivate1.UserCad.CreateRectangle(-80,44,-79,45);
  MapTivate1.UserCad.SaveToFile('MyCad01');
  MapTivate1.RedrawMap;
end;
```

Styles[Index:Integer]:ITRStyle

Indexed array containing all the styles [0..255] of the UserCAD theme.

Sample Code (Delphi 5)

```
procedure TForm1.CADStyleClick(Sender: TObject);
```

```
begin
  With MapTivate1.UserCAD.styles[0] do
  begin
    Pen.color:=clred;
    Pen.width:=4;
    //-
    Brush.color:=clyellow;
    Brush.Style:=4;
  end;
end;
```

Upper:Double

Upper limit of visibility for the UserCAD Theme.

Visible:Boolean

Determines whether the UserCAD theme will be visible.

13. ITRPaint Object (MapTivate.UserPaint)

This object is similar to a theme object, but gives the user the ability to draw transient primitives (points, lines, ellipses, etc.) on the control surface, at will. ***As the last operation, the user must call RefreshMap to cause all transient objects to become visible.***

Brush:ITRBrush

Sets the brush properties to be used for polygons drawn by the IUserPaint interface.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  with MapTivate1 do
  begin
    UserPaint.Mark.Size:=20;
    UserPaint.Mark.Color:=clblue;
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    UserPaint.Brush.style:=3;
    UserPaint.Brush.color:=clYellow;
    UserPaint.Font.height:=-18;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clyellow;
    Userpaint.smRectangle(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

SmCircle(X,Y:Double; R:Double; Aspect:Double)

Draw circle using parameters specified by the user. Note that a horizontal or vertical ellipse may be drawn by varying the Aspect parameter.

X,Y Center Coordinates (lon/Lat)
R Radius (in degrees)
Aspect Ratio of Vertical to horizontal diameter (1.0 for circle)

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    UserPaint.Mark.Size:=20;
    UserPaint.Mark.Color:=clblue;
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    UserPaint.Brush.style:=3;
    UserPaint.Brush.color:=clYellow;
    UserPaint.Font.height:=-18;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clyellow;
    Userpaint.smRectangle(MapLeft,MapBottom,MapRight,MapTop);
    // Set Aspect Ratio for Vert Ellipse
    AspRatio:=2;
    Userpaint.smCircle(MapCenterX+3,MapCenterY+3,3,2);
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    // Change color and width to draw line
      UserPaint.pen.width:=4;
      UserPaint.pen.color:=clblue;
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command33_Click()
  ' Set some line attributes
  With MapTivate1.UserPaint.Pen
    .Color = vbBlue
    .BackColor = vbRed
    .Width = 3
    .OuterWidth = 5
  End With
  ' Draw a transient Line from top left to bottom right
  MapTivate1.UserPaint.SmLine MapTivate1.MapLeft, MapTivate1.MapTop, MapTivate1.MapRight,
MapTivate1.MapBottom
  ' Set some brush properties
  With MapTivate1.UserPaint.Brush
    .BackColor = vbRed
     ' Grid Pattern
    .Style = 6
    .Color = vbGreen
    .Transparent = True
  End With
  ' Draw Circle at Center point
  With MapTivate1
    .UserPaint.SmCircle .MapCenterX, .MapCenterY, .ZoomScale / 30, 1
  End With
  ' Refresh Map to show line
```

```
  MapTivate1.RefreshMap
End Sub
```

SmDisk(X,Y:Double, R:Integer)

Draws a Shadowed disk with center at the specified **X,Y** coordinates (degrees) and with a Radius on **R** pixels.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  with MapTivate1 do
  begin
    Userpaint.smDisk (MapCenterX-3, MapCenterY,20);
   ReFreshMap;
             End;
```

Sample Code (Visual Basic)

```
Private Sub Command33_Click()
  ' Set some line attributes
  With MapTivate1.UserPaint.Pen
    .Color = vbBlue
    .BackColor = vbRed
    .Width = 3
    .OuterWidth = 5
  End With
  ' Draw a transient Line from top left to bottom right
  MapTivate1.UserPaint.SmLine MapTivate1.MapLeft, MapTivate1.MapTop, MapTivate1.MapRight,
MapTivate1.MapBottom
  ' Set some brush properties
  With MapTivate1.UserPaint.Brush
    .BackColor = vbRed
     ' Grid Pattern
    .Style = 6
    .Color = vbGreen
    .Transparent = True
  End With
   ' Draw a Disk object
  With MapTivate1
    .UserPaint.SmDisk .MapCenterX - .ZoomScale / 40, .MapCenterY - .ZoomScale / 40, 100
  End With
  ' Refresh Map to show line
  MapTivate1.RefreshMap
End Sub
```

Font:ITRFont

Sets the Font properties to be used by the IUserPaint interface.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  with MapTivate1 do
  begin
    UserPaint.Mark.Size:=20;
    UserPaint.Mark.Color:=clblue;
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    UserPaint.Brush.style:=3;
```

```
    UserPaint.Brush.color:=clYellow;
    UserPaint.Font.height:=-18;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clyellow;
    Userpaint.smRectangle(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

SmLine(X1,Y1,X2,Y2:Double)

Draw a line between points specified by the user.

X1,Y1 Starting point coordinates (Lon/Lat)
X2,Y2 End point coordinates (Lon/Lat)

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  with MapTivate1 do
  begin
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    Userpaint.smRectangle(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    // Change color and width to draw line
      UserPaint.pen.width:=4;
      UserPaint.pen.color:=clblue;
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command33_Click()
  ' Set some line attributes
  With MapTivate1.UserPaint.Pen
    .Color = vbBlue
    .BackColor = vbRed
    .Width = 3
    .OuterWidth = 5
  End With
  ' Draw a transient Line from top left to bottom right
  MapTivate1.UserPaint.SmLine MapTivate1.MapLeft, MapTivate1.MapTop, MapTivate1.MapRight,
MapTivate1.MapBottom
  ' Set some brush properties
  With MapTivate1.UserPaint.Brush
    .BackColor = vbRed
     ' Grid Pattern
    .Style = 6
    .Color = vbGreen
    .Transparent = True
  End With
  ' Refresh Map to show line
  MapTivate1.RefreshMap
End Sub
```

Mark:ITRMark

Sets the Mark properties to be used by the ITRUserPaint interface when drawing point objects. See ITRMArk Object section for details.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  with MapTivate1 do
  begin
    UserPaint.Mark.Size:=20;
    UserPaint.Mark.Color:=clblue;
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command34_Click()
' Zoom Out/In for appropriate scale
  MapTivate1.ZoomScale = 100
  With MapTivate1
    ' Set Mapr Properties
    .UserPaint.Mark.Size = 20
    .UserPaint.Mark.Color = vbBlue
    ' Set Pen Properties
    .UserPaint.Pen.Width = 2
    .UserPaint.Pen.Color = vbblack
    ' Set Font properties
    .UserPaint.Font.Color = vbgreen
    .UserPaint.Font.Height = -18
    .UserPaint.Font.Style = 3
    .UserPaint.SmPoint .MapCenterX - 1, .MapCenterY + 1
    ' Text at the viewport centerpoint
    .UserPaint.SmText .MapCenterX - 1.001, .MapCenterY + 0.8, "Sample UserDraw Text"
  End With
  MapTivate1.RefreshMap
End Sub
```

Pen:ITRPen

Sets the Pen properties to be used by the IUserPaint interface when drawing line objects. See the ITRPen Object section for details.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  with MapTivate1 do
  begin
    UserPaint.Mark.Size:=20;
    UserPaint.Mark.Color:=clblue;
```

```
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    UserPaint.Brush.style:=3;
    UserPaint.Brush.color:=clYellow;
    UserPaint.Font.height:=-18;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clyellow;
    Userpaint.smRectangle(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

SmPoint(X,Y: Double);

Place a marker (point) at the user-specified coordinates, using the current mark style.

X,Y Point coordinates (Lon/Lat)

Sample Code (Delphi 5)

```
procedure TForm1.UserPaint1Click(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    // Set Mark Properties
    UserPaint.Mark.Style:=strtoint(edit9.text);
    UserPaint.Mark.SymbolFont:='Symbols';
    UserPaint.Mark.Symbol:=strtoint(edit10.text);
    UserPaint.Mark.Size:=20;
    UserPaint.Mark.Color:=clblue;
    // Set Pen Properties
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    // Set Brush Properties
    UserPaint.Brush.style:=3;
    UserPaint.Brush.BackColor:=clwhite;
    UserPaint.Brush.color:=clYellow;
    // Set Font Properties
    UserPaint.Font.height:=-10;
    UserPaint.Font.color:=clblue;
    UserPaint.Font.BackColor:=clred;
    UserPaint.Font.Alignment:=strtoint(edit4.text);
    UserPaint.Font.style:=strtoint(edit11.text);
    // Draw Rectangle
    Userpaint.smRectangle(MapLeft,MapBottom,MapRight,MapTop);
    // Set Aspect Ratio and draw Vert Ellipse
    AspRatio:=2;
    Userpaint.smCircle(MapCenterX+3,MapCenterY+3,3,2);
    // Draw a few points and identify them with text
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    Userpaint.smtext(MapCenterX-1,MapCenterY+1,'(-1,1)');
    Userpaint.smPoint(MapCenterX-2,MapCenterY+1);
    Userpaint.smtext(MapCenterX-2,MapCenterY+1,'(-2,1)');
    Userpaint.smPoint(MapCenterX-2,MapCenterY+2);
    Userpaint.smtext(MapCenterX-2,MapCenterY+2,'(-2,2)');
    Userpaint.smPoint(MapCenterX-1,MapCenterY+2);
    Userpaint.smtext(MapCenterX-1,MapCenterY+2,'(-1,2)');
    // Change color and width to draw line
    UserPaint.pen.width:=4;
    UserPaint.pen.color:=clblue;
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
```

```
    // Set Larger Font and color for Label
    UserPaint.Font.color:=clgreen;
    UserPaint.Font.BackColor:=clblue;
    UserPaint.Font.height:=-18;
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Main Sample Text');
    RefreshMap;
  end;
end;
```

SmPolyPoint(Var Points:TRPoint; Var Counts:Integer; N:Integer)

Creates a Multi-point point object, as defined by the user, based on a variable points array.

PPt: TRPoints array
PPn: Counts array
N: Number of elements (points) in the object.

Sample Code (Delphi 5)

```
procedure TForm1.PolyPointClick(Sender: TObject);
var points:array of trpoint;
    counts:array of integer;
begin
  MapTivate1.ZoomCenter(-72,42);
  MapTivate1.smscale:=100;
  // Use Delphi's SetLength which
  // Sets the length of a string or dynamic-array variable
  setlength(points,10);
  setlength(counts,1);
  points[0].x:=-70;  points[0].y:=44;
  points[1].x:=-72;  points[1].y:=42;
  points[2].x:=-74;  points[2].y:=44;
  points[3].x:=-72;  points[3].y:=44;
  // Set the # points in object to 4
  counts[0]:=4;

  // Decide what to use for markers
  PolyPtMarkOpt:=not(PolyPtMarkOpt);
  If PolyPtMarkOpt=true then
  begin
    // Set Mark ptoperties
    MapTivate1.UserPaint.Mark.Style:=5;
    MapTivate1.UserPaint.Mark.Size:=20;
    MapTivate1.UserPaint.Mark.BorderColor:=clYellow;
    MapTivate1.UserPaint.Mark.Color:=clblue;
  end else
  begin
    // or. alternatively, use a user bitmap
    MapTivate1.UserPaint.Mark.style:=-1;
    MapTivate1.UserPaint.Mark.Bitmap.LoadImage('redbtn.bmp');
  end;

  // Create one multipoint object
  MapTivate1.Userpaint.smPolyPoint(points[0] , counts[0] ,1);
  MapTivate1.RefreshMap;
end;
```

SmPolyLine(Var Points:TRPoint; Var Counts:Integer; N:Integer)

Creates a Multi-segment Polyline object, as defined by the user, based on a variable points array.

PPt: TRPoints array
PPn: Counts Array
N: Number of elements (line segments) in the object

Sample Code (Delphi 5)

```
procedure TForm1.PolyLineClick(Sender: TObject);
var points:array of trpoint;
    counts:array of integer;
    NumObjects:integer;
begin
  MapTivate1.ZoomCenter(-72,42);
  MapTivate1.smscale:=500;
  // Use Delphi's SetLength which
  // Sets the length of a string or dynamic-array variable
  setlength(points,10);
  /// will create up to 5 objects
  setlength(counts,2);
  points[0].x:=-70;  points[0].y:=44;
  points[1].x:=-72;  points[1].y:=42;
  points[2].x:=-74;  points[2].y:=44;
  points[3].x:=-70;  points[3].y:=43;
  points[4].x:=-72;  points[4].y:=41;
  points[5].x:=-74;  points[5].y:=43;
  points[6].x:=-74;  points[6].y:=40;
  // Will create two Polyline objects from the points array
  NumObjects:=2;
  // The first object will contain 3 points Points[0]..Points[2]
  counts[0]:=3;
  // The second object will contain 4 points Points[3]..Points[6]
  counts[1]:=4;
  MapTivate1.UserPaint.pen.color:=clblue;
  MapTivate1.UserPaint.pen.width:=4;
  MapTivate1.Userpaint.smPolyLine(points[0] , counts[0] ,NumObjects);
  MapTivate1.RefreshMap;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command35_Click()
Dim mpoints(0 To 10) As TRPoint
Dim mcounts(0 To 2) As Long, NumObjects As Long
  MapTivate1.ZoomCenter -72, 42
  MapTivate1.ZoomScale = 500
  mpoints(0).X = -70
  mpoints(0).Y = 44
  mpoints(1).X = -72
  mpoints(1).Y = 42
  mpoints(2).X = -74
  mpoints(2).Y = 44
  mpoints(3).X = -70
  mpoints(3).Y = 43
  mpoints(4).X = -72
  mpoints(4).Y = 41
  mpoints(5).X = -74
  mpoints(5).Y = 43
  mpoints(6).X = -74
  mpoints(6).Y = 40
  'Will create two Polyline objects from the points array
  NumObjects = 2
  'The first object will contain 3 points Points[0]..Points[2]
  mcounts(0) = 3
  'The second object will contain 4 points Points[3]..Points[6]
  mcounts(1) = 4
  MapTivate1.UserPaint.Pen.Color = vbBlue
  MapTivate1.UserPaint.Pen.Width = 4
  MapTivate1.UserPaint.SmPolyline mpoints(0), mcounts(0), NumObjects
```

```
  MapTivate1.RefreshMap
End Sub
```

SmPolygon(Var Points:TRPoint; Var Counts:Integer; N:Integer)

Creates a Multipart Polygon object, as defined by the user, based on a variable points array. It allows the user to draw a number of discrete polygons that are part of the same object.

PPt: TRPoints array
PPn: Counter array
N: Number of elements (pogygons) in the object

Sample Code (Delphi 5)

```
procedure TForm1.PolyGonClick(Sender: TObject);
var points:array of trpoint;
    counts:array of integer;
begin
  MapTivate1.ZoomCenter(-72,42);
  MapTivate1.smscale:=300;
  // Use Delphi's SetLength which
  // Sets the length of a string or dynamic-array variable
  setlength(points,15);
  setlength(counts,3);
  points[0].x:=-70;  points[0].y:=44;
  points[1].x:=-72;  points[1].y:=42;
  points[2].x:=-74;  points[2].y:=44;

  points[3].x:=-74;  points[3].y:=40;
  points[4].x:=-72;  points[4].y:=44;
  points[5].x:=-70;  points[5].y:=40;

  points[6].x:=-68;  points[6].y:=46;
  points[7].x:=-68;  points[7].y:=38;
  points[8].x:=-76;  points[8].y:=38;
  points[9].x:=-76;  points[9].y:=46;
  points[10].x:=-68;  points[10].y:=46;

  counts[0]:=3;
  counts[1]:=3;
  counts[2]:=5;

  MapTivate1.UserPaint.pen.color:=clblue;
  MapTivate1.UserPaint.Pen.width:=4;
  MapTivate1.UserPaint.brush.color:=clred;
  MapTivate1.UserPaint.brush.style:=integer(bssolid);
  // Note that based on the way Windows treats overlapping bruches...
  // First poly in object is painted with solid red
  // Second poly is painted in solid red, areas overlaping the
  // first polygon are clear (hollow).
  // Third poly is painted solid read.  Areas overlapping either
  // poly #1 or poly #2 are clear.  Areas overlapping sections common
  // to poly #1 and poly #2 (which were clear above), are painted solid...
  MapTivate1.Userpaint.smPolygon(points[0] , counts[0] ,3);
  MapTivate1.RefreshMap;
end;
```

SmRectangle(X1,Y1,X2,Y2:Double)

Draw a rectangle between the two points specified by the user.

X**1,Y1** Top Left corner of rectangle (Lon/Lat)
X2,Y2 Bottom Right corner of rectangle (Lon/Lat)

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  with MapTivate1 do
  begin
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    Userpaint.smRectangle(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    // Change color and width to draw line
      UserPaint.pen.width:=4;
      UserPaint.pen.color:=clblue;
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

SmRing(X,Y:Double, R,Delta:Integer)

Draws a 3D ring with center at X,Y (Lon/Lat), Outer Radius **R** (pixels) and annulus width **Delta** (pixels).

Sample Code (Delphi 5)

```
procedure TForm1.MapPaintNewClick(Sender: TObject);
Var AspRatio:double;
begin
  with MapTivate1 do
  begin
    // Set Pen Properties
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    UserPaint.pen.outerwidth:=9;
    UserPaint.pen.backcolor:=clgreen;
    // Set Brush Properties
    UserPaint.Brush.style:=3;
    UserPaint.Brush.BackColor:=clwhite;
    UserPaint.Brush.color:=clYellow;
    // Draw Ring
    // OterRadius = 90 pixels
    // Delta Radius (NOT Inner) = 20 pixels
   Userpaint.smRing(MapCenterX, MapCenterY,190,20);
  MapTivate1.Refresh;
End;
```

Sample Code (Visual Basic)

```
Private Sub Command39_Click()
  With MapTivate1
   ' Set Pen Properties
   .UserPaint.Pen.Width = 2
   .UserPaint.Pen.Color = vbBlack
   .UserPaint.Pen.OuterWidth = 9
   .UserPaint.Pen.BackColor = vbGreen
   ' Set Brush Properties
   .UserPaint.Brush.Style = 3
   .UserPaint.Brush.BackColor = vbWhite
   .UserPaint.Brush.Color = vbYellow
```

```
    ' Draw Ring
    ' OterRadius = 90 pixels
    ' Delta Radius (NOT Inner) = 20 pixels
    .UserPaint.SmRing MapCenterX, MapCenterY, 190, 20
    .RefreshMap
  End With
End Sub
```

SmSphere(X,Y:Double; Radius:Integer)

Draws a 3D shere with highlighting with center at X,Y (Lat/Lon) and the specified radius in pixels.

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  // Paint 3D sphere on themap
  Userpaint.smSphere(MapCenterX+3, MapCenterY,60);
  MapTivate1.Refresh;
End;
```

SmText(X,Y:Double; S:String)

Place text at the coordinates specified by the user. Note that Font.Alignment controls where in the text bounding rectangle the text is placed at.

X,Y Coordinates (Lon/Lat) for placing the text
S Text to be placed at X,Y

Sample Code (Delphi 5)

```
procedure TForm1.UserPaintClick(Sender: TObject);
begin
  with MapTivate1 do
  begin
    UserPaint.pen.width:=2;
    UserPaint.pen.color:=clblack;
    Userpaint.smRectangle(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smPoint(MapCenterX-1,MapCenterY+1);
    // Change color and width to draw line
      UserPaint.pen.width:=4;
      UserPaint.pen.color:=clblue;
    Userpaint.smLine(MapLeft,MapBottom,MapRight,MapTop);
    Userpaint.smtext(MapCenterX+1,MapCenterY-1,'Sample Text');
    RefreshMap;
  end;
end;
```

14. TRDrawObject

Defines the items that are managed by the UserDraw specialized theme.

BW:Boolean

If this flag is set to tru, then the User Object is set to gray (useful for denoting marked/selected objects).

Caption:String

The caption to be printed below the user item.

Sample Code (Delphi 5)

```
procedure TForm1.UserDrawObjClick(Sender: TObject);
var n,m:integer;
   U:TRUserObject;
    bm:tbitmap;
    d:trect;
      UL:TRUserDraw;
    pt:trpoint;
      UsrLr:TRUserDraw;
    X,y:double;

begin
  bm:=tbitmap.create;
  bm.width:=image3.picture.bitmap.width;
  bm.height:=image3.picture.bitmap.height;
  setrect(d,0,0,bm.width,bm.height);
  bm.handletype:=bmDIB;
  bm.pixelformat:=pf24bit;
  bm.Canvas.copyrect(d,image3.picture.bitmap.canvas,d);
  Paintbox2.Canvas.copyrect(d,bm.canvas,d);

  UsrLr:=MapTivate1.UserDraw;
  USrLr.Styles[0].Font.color:=clred;
  UsrLr.Styles[0].Font.alignment:=3;
  Usrlr.Styles[2].font.color:=clblue;

  u:=UsrLr.NewObject;
  u.style:=0;
  begin
    U.Caption:='Sample1';
    U.x:=-80;
    U.y:=45;
    // Use handle to the bitmap already loaded to the Image box
    Usrlr.styles[0].mark.bitmap.handle:=bm.handle;
    UsrLr.styles[0].Mark.bitmap.angle:=10;
  end;

  u:=UsrLr.NewObject;
  u.style:=2;
  begin
    U.Caption:='Sample2';
    U.x:=-80;
    U.y:=47;
    // Load Image from File
   UsrLr.styles[2].Mark.bitmap.LoadImage('bin.bmp');
   UsrLr.styles[2].Mark.bitmap.angle:=5;
```

```
  end;
  MapTivate1.ZoomCenter(-80,46);

  Panel2.font.color:=clblack;
  UsrLr.upper:=5000;
  UsrLr.Styles[0].Mark.upper:=5000;
  MapTivate1.redrawmap;
  MessageBeep(0);
  Panel2.caption:='# Items[b] = '+inttostr(MapTivate1.UserDraw.count);
  with MapTivate1 do
  begin
    ZoomMapRect(UserDraw.Extents.xmin,UserDraw.Extents.ymin,
                UserDraw.Extents.xmax, UserDraw.Extents.ymax);
  end;
end;
```

Enabled:Boolean

Defines whether the User Item will be enabled.

ID:LongInt

Simple tag to allow the user to possibly assign unique ID to this object.

Index:Integer

The position of the User Item in the User Items list.

Lower:Double

Lower Visibility scale for this user item.

Sample Code (Delphi 5)

```
procedure TForm1.ChgUserObjClick(Sender: TObject);
begin
  // Change the upper/lower vis thresholds
  UsrLr.Objects[0].upper:=70;
  UsrLr.Objects[0].lower:=0.1;
  UsrLr.Objects[1].upper:=35;
  UsrLr.Objects[1].lower:=1;
  MapTivate1.RefreshMap;
end;
```

SecCaption:String

Parallel to Caption, but used in combination with SecStyle, allows the user to place text independent of the actual object, i.e., Caption is controlled by Style, while SecCaption is controlled by SecStyle.

SecStyle:Integer

Parallel to Style, but used in combination with SecCaption, allows the user to place text independent of the actual object, i.e., Caption is controlled by Style, while SecCaption is controlled by SecStyle.

Selected:Boolean

Defines if a given object is currently selected, or not.

Sample Code (Delphi 5)

```
procedure TForm1.SelAllUserClick(Sender: TObject);
var i:integer;
begin
  For i:=0 to MapTivate1.USerDraw.Count-1 do
  begin
    If MapTivate1.USerDraw.Objects[i].Selected= true then
            ListBox1.Items.Add('Item # '+inttostr(i)+' is Selected');
  end;
end;
```

Style:Integer

The style # that this User Item is assigned to.

Sample Code (Delphi 5)

```
procedure TForm1.SUserClick(Sender: TObject);
 // Set the style of object(0) to that of object(2)
  UsrLr.Objects[0].style:=UsrLr.Objects[1].style;
  MapTivate1.RefreshMap;
end;
```

Upper:Double

Upper visibility scale for this user item.

Sample Code (Delphi 5)

```
procedure TForm1.ChgUserObjClick(Sender: TObject);
begin
  // Change the upper/lower vis thresholds
  UsrLr.Objects[0].upper:=70;
  UsrLr.Objects[0].lower:=0.1;
  UsrLr.Objects[1].upper:=35;
  UsrLr.Objects[1].lower:=1;
  MapTivate1.RefreshMap;
end;
```

Visible:Boolean

Defines whether the User Item will be visible.

X:Double

The X-coordinate (Longitude) of the center of the user object.

Sample Code (Delphi 5)

```
procedure TForm1.ChgUserObjClick(Sender: TObject);
begin
  // Change the upper vis threshold
  UsrLr.Objects[0].upper:=70;
   // Move object to new coordinates
  ListBox3.Items.add('XYc: '+floattostr(UsrLr.Objects[0].x)
                    +', '+floattostr(UsrLr.Objects[0].y));
  UsrLr.Objects[0].x:=UsrLr.Objects[0].x*1.01;
  UsrLr.Objects[0].y:=UsrLr.Objects[0].y*0.99;
  MapTivate1.RefreshMap;
  ListBox3.Items.add('New: '+floattostr(UsrLr.Objects[0].x)
                    +', '+floattostr(UsrLr.Objects[0].y));
end;
```

Y:Double

The Y-coordinate (Latitude) of the center of the user object.

Sample Code (Delphi 5)

```
procedure TForm1.ChgUserObjClick(Sender: TObject);
begin
  // Change the upper vis threshold
  UsrLr.Objects[0].upper:=70;
   // Move object to new coordinates
  ListBox3.Items.add('XYc: '+floattostr(UsrLr.Objects[0].x)
                    +', '+floattostr(UsrLr.Objects[0].y));
  UsrLr.Objects[0].x:=UsrLr.Objects[0].x*1.01;
  UsrLr.Objects[0].y:=UsrLr.Objects[0].y*0.99;
  MapTivate1.RefreshMap;
  ListBox3.Items.add('New: '+floattostr(UsrLr.Objects[0].x)
                    +', '+floattostr(UsrLr.Objects[0].y));
end;
```

15. TRDraw Object (MapTivate.UserDraw)

A specialized type of theme for manipulating discrete bitmaps, e.g., moving icons for a GPS interface. Each item may be assigned to one of 255 styles.

AddStyle():Integer;

Adds a style a style to the list of TRDraw Object defined styles (appends it to the end of the list) and increments *StyleCount* by one. Note that a Style needs to be added before its attributes can be modified. The only exception is Style[0], which is the default style and is pre-defined.

Clear()

Clears all items on the user drawing layer and releases all resources.

Sample Code (Delphi 5)

```
procedure TForm1.USerDgw2Click(Sender: TObject);
begin
  //Clear The User drawing layer
  MapTivate1.UserDraw.clear;
end;
```

ClearStyles()

Clears all currently defined styles in the UserDraw Object.

Count:Integer

Total number of bitmap items in UserDraw.

Sample Code (Delphi 5)

```
procedure TForm1.USerDgw2Click(Sender: TObject);
begin
  // Display # of objects on user layer
  Panel2.caption:='# Items = '+inttostr(MapTivate1.UserDraw.count);
  //Clear The User drawing layer
  MapTivate1.UserDraw.clear;
end;
```

Delete(Index,Num:Integer)

Delete **Num** user items starting with **Index**.

Sample Code (Delphi 5)

```
procedure TForm1.USerDgw2Click(Sender: TObject);
begin
  // Display # of objects on user layer
  Panel2.caption:='# Items[a] = '+inttostr(MapTivate1.UserDraw.count);
  // Delete the first USer Item
  MapTivate1.UserDraw.Delete(0,1);
  // Re-Display the adjusted # of items
  Panel3.caption:='# Items[b] = '+inttostr(MapTivate1.UserDraw.count);
end;
```

DeleteStyle(N:Integer)

Deletes the specified Style from the UserDraw Object.

Sample Code (Delphi 5)

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  // Delete Style # 5
  MapTivate1.USerDraw.DeleteStyle(5);
  // Clear Current Styles
  MapTivate1. UserDraw.ClearStyles;
end;
```

Enabled:Boolean

Defines if the UserDraw object is enabled.

Extents:TRExtents

Extents of all the user objects currently in UserDraw.

Sample Code (Delphi 5)

```
procedure TForm1.UsrDwgXtentsClick(Sender: TObject);
var TpExt:TRExtents;
begin
  TpExt:=MapTivate1.UserDraw.Extents;
  // Display current Extents
  Label2.caption:=floattostr(TpExt.xmin);
  Label3.caption:=floattostr(TpExt.ymin);
  Label4.caption:=floattostr(TpExt.xmax);
  Label5.caption:=floattostr(TpExt.ymax);
  MessageBeep(0);
end;
```

FindObjectAtPoint(X,Y:Double);

Locates all User Objects within 8 pixels (default value of Tolerance) from the specified coordinates and fires a FindUserObject event for each object found.

Sample Code (Delphi 5)

```
procedure TForm1.FindPoinAtUSrClick(Sender: TObject);
begin
  // Find USer Objects at two different locations
  // And populate the list box with the info of
  // the user objects.
  MapTivate1.UserDraw.FindObjectAtPoint(-80,47);
  MapTivate1.UserDraw.FindObjectAtPoint(-80,45);
end;

procedure TForm1.MapTivate1FindUserObject(Sender: TObject;
  const value: ITRUserObject);
var s:String;
begin
  s:=floattostr(value.x)+', '+floattostr(value.y);
  Listbox3.Items.add('Found Object #'+inttostr(Value.index)+', Located at: '+s);
end;
```

FindObjectInRect(X1,Y1,X2,Y2:Double);

Locates all User Objects within the rectangle specified by the two points, and fires a FindUserObject event for each object found.

Sample Code (Delphi 5)

```
procedure TForm1.FindPoinAtUSrClick(Sender: TObject);
begin
  // Find USer Objects in Rectangle
  MapTivate1.UserDraw.FindObjectInRect(-70,40,90,47);
end;

procedure TForm1.MapTivate1FindUserObject(Sender: TObject;
  const value: ITRUserObject);
var s:String;
begin
  s:=floattostr(value.x)+', '+floattostr(value.y);
  Listbox3.Items.add('Found Object #'+inttostr(Value.index)+', Located at: '+s);
end;
```

InsertStyle(N:LongInt);

Inserts a style immediately following the style ‘N’ specified by the user.

Objects[Index:Integer]:ITRDrawObject

Index list of all items on the user theme.

Sample Code (Delphi 5)

```
procedure TForm1.USerDgw2Click(Sender: TObject);
begin
  Panel3.caption:=FloatToStr(MapTivate1.UserDraw.Objects[0].x)+', '
                  +FloatToStr(MapTivate1.UserDraw.Objects[0].y);
  Panel5.caption:=FloatToStr(MapTivate1.UserDraw.Objects[1].x)+', '
                  +FloatToStr(MapTivate1.UserDraw.Objects[1].y);
end;
```

LoadFromFile(FileName:String);

Loads a UserDraw definition from a file specified by the user. See Appendix “B” for a sample listing of a UserDraw file.

Sample Code (Delphi 5)

```
procedure TForm1.DwgLoadFromFileClick(Sender: TObject);
begin
  // Load User items
  MapTivate1.UserDraw.LoadFromFile('MyOwnObjects');
end;
```

Lower:Double

Lower limit of visibility for the User Item Theme.

NewObject():TRUserObject

Creates a new object position in the user item list.

Sample Code (Delphi 5)

```
procedure TForm1.UserDrawObjClick(Sender: TObject);
var U:TRUserObject;
    n,m:integer;
    bm:tbitmap;
    d:trect;
    UL:TRUserDraw;
    UsrLr:TRUserDraw;
Begin
  bm:=tbitmap.create;
  bm.pixelformat:=pf24bit;
  bm.width:=image3.picture.bitmap.width;
  bm.height:=image3.picture.bitmap.height;
  setrect(d,0,0,bm.width,bm.height);

  bm.Canvas.copyrect(d,image3.picture.bitmap.canvas,d);
// Add two styles to use further down
  UsrLr.AddStyle; // Adding style[1]
  UsrLr.AddStyle;  // Adding Style[0]
  UsrLr:=MapTivate1.UserDraw;
  USrLr.Styles[0].Font.color:=clred;
  UsrLr.Styles[0].Font.alignment:=3;
  Usrlr.Styles[2].font.color:=clblue;

  u:=UsrLr.NewObject;
  u.style:=0;
  begin
    U.Caption:='Sample1';
    U.x:=-80;
    U.y:=45;
    // Usrlr.styles[0].mark.bitmap.handle:=bm.handle;
    UsrLr.styles[0].Mark.bitmap.LoadImage('redbtn.bmp');
  end;

  u:=UsrLr.NewObject;
  u.style:=2;
  begin
    U.Caption:='Sample2';
    U.x:=-80;
    U.y:=47;
    Usrlr.styles[2].mark.bitmap.handle:=bm.handle;
  end;

  Panel2.font.color:=clblack;
  UsrLr.upper:=5000;
  UsrLr.Styles[0].Mark.upper:=5000;
  MapTivate1.redrawmap;
  MessageBeep(0);
  bm.free;
  Panel2.caption:='# Items[b] = '+inttostr(MapTivate1.UserDraw.count);
  with MapTivate1 do
  begin
    ZoomMapRect(UserDraw.Extents.xmin,UserDraw.Extents.ymin,
                UserDraw.Extents.xmax, UserDraw.Extents.ymax);
  end;
end;
```

Sample Code (Visual Basic)

```
Private Sub Command10_Click()
' Animate a user object using a number of pre-defined points.
 Dim chkEnd As Double
 Dim bm As PictureBox
 Dim pt As TRPoint, X As Double, Y As Double
   Set TempCheckTheme = MapTivate1.Themes("County-Roads")
   If ObjPtr(TempCheckTheme) = 0 Then
     ' Pop Dialog and notify user
     UserResponse = MsgBox("Problem: County-Roads Theme is NOT connected!", vbOKOnly,
"Error was encountered!")
   Else
     ChDrive App.Path
     ChDir App.Path
     chkEnd = 0
     CommonDialog1.DialogTitle = "Select Points file to Open for Animation..."
     CommonDialog1.Filter = "Point Files (*.pts)|*.pts|All Files|*.*"
     CommonDialog1.ShowOpen
     PtsFile = CommonDialog1.FileName
     If Len(PtsFile) > 0 Then
       Open PtsFile For Input As #1
       i = 1
       ' Repeat until the simple EOF marker is read
       While chkEnd <> 99999999
         Input #1, AnimPoints(i).X, AnimPoints(i).Y
         chkEnd = AnimPoints(i).X
         i = i + 1
       Wend
       Close #1
       PcPt = i - 2
       Text1.Text = "Loaded: " + Str(PcPt) + " Points"
       Set UsrLr = MapTivate1.UserDraw
       Set u = UsrLr.NewObject
       u.Style = 0
       u.X = AnimPoints(1).X
       u.Y = AnimPoints(1).Y
       ' Use handle to the bitmap already loaded to the Image box
       UsrLr.Styles(0).Mark.Bitmap.Handle = Image1.Picture.Handle
       UsrLr.Styles(0).Mark.Bitmap.Transparent = True
       AnimationMode = True
       AnimInterval = Val(Text2.Text)
       If AnimInterval < 1 Or AnimInterval > 5 Then AnimInterval = 1
       MapTivate1.ZoomScale = 2
       ' Zoom in to the first point
       MapTivate1.ZoomCenter u.X + 0.02, u.Y
       Ustart = 0
       Timer1.Interval = 50
       Timer1.Enabled = True
     Else
     End If
   End If
End Sub
```

SaveToFile(Filename:String)

Saves the Current UserDraw definition to a user-specified file. See Appendix “B” for a sample listing of a UserDraw file. Unless a file extension is explicitly specified, the extension **.TRU** is added.

Sample Code (Delphi 5)

```
procedure TForm1.DwgSaveToFileClick(Sender: TObject);
begin
   // Save user item definition to file
   MapTivate1.UserDraw.SaveToFile('MyOwnObjects');
   // Clear USer Items
   MapTivate1.UserDraw.Clear;
```

```
end;
```

SelectAll(Option:Integer)

Select (or DeSelects) all User objects for a subsequent operation. N=0 Unselect, N=1 Select, N=2 Toggle current selection status. Note that Selected Status will not be visible until RefreshMap or RedrawMap is called.

Sample Code (Delphi 5)

```
procedure TForm1.SelAllUserClick(Sender: TObject);
begin
  // Select All objects
  MapTivate1.UserDraw.SelectAll(1);
  // Delete All selected objects and update the map
  MapTivate1.UserDraw.SelectDelete;
  MapTivate1.RedrawMap;
end;
```

SelectDelete()

Delete currently selected User Objects.

Sample Code (Delphi 5)

```
procedure TForm1.SelAllUserClick(Sender: TObject);
begin
  // Select All objects
  MapTivate1.UserDraw.SelectAll(1);
  // Delete All selected objects and update the map
  MapTivate1.UserDraw.SelectDelete;
  MapTivate1.RedrawMap;
end;
```

SelectPoint(x,y:Double; Option:Integer)

Select (or DeSelect) User objects within 8 pixels (default tolerance) of the specified X,Y coordinates, for a subsequent operation. Option=0 Unselect, Option =1 Select, Option =2 Toggle current selection status.

Sample Code (Delphi 5)

```
procedure TForm1.SelRecUSerClick(Sender: TObject);
begin
   MapTivate1.GetCoords(Xo,Yo);
    MapTivate1.UserDraw.SelectPoint(Xo,Yo,1);
End;
```

SelectRange(Index, Num:Integer; Option:Integer)

Select (or DeSelects) Num User objects starting with Index, for a subsequent operation. Option=0 Unselect, Option =1 Select, Option =2 Toggle current selection status.

Sample Code (Delphi 5)

```
procedure TForm1.SelRecUSerClick(Sender: TObject);
begin
  // Select objects
  MapTivate1.UserDraw.SelectRange(0,1,1);
  // Delete All selected objects and update the map
  MapTivate1.UserDraw.SelectDelete;
  MapTivate1.RedrawMap;
end;
```

SelectRectangle(X1,Y1:Double; X2,Y2:Double; Option:integer)

Select (or DeSelects) User objects inside the specified rectangle, for a subsequent operation. N=0 Unselect, N=1 Select, N=2 Toggle current selection status.

Sample Code (Delphi 5)

```
procedure TForm1.SelRecUSerClick(Sender: TObject);
begin
  // Select objects in specified rectangle
  MapTivate1.UserDraw.SelectRectangle(-160,32,-60,50,1);
  // Delete All selected objects and update the map
  MapTivate1.UserDraw.SelectDelete;
  MapTivate1.RedrawMap;
end;
```

Styles[Index:Integer]:ITRStyle

Indexed List of Styles to be used for the User Items Theme. User Items may be assigned to one of 255 styles, defined by the user.

Sample Code (Delphi 5)

```
procedure TForm1.UserDrawObjClick(Sender: TObject);
var U:TRUserObject;
    n,m:integer;
    bm:tbitmap;
    d:trect;
    UL:TRUserDraw;
    pt:trpoint;
    UsrLr:TRUserDraw;

begin
  bm:=tbitmap.create;
  bm.width:=image3.picture.bitmap.width;
  bm.height:=image3.picture.bitmap.height;
  setrect(d,0,0,bm.width,bm.height);

  bm.Canvas.copyrect(d,image3.picture.bitmap.canvas,d);

  UsrLr:=MapTivate1.UserDraw;
  USrLr.Styles[0].Font.color:=clred;
  UsrLr.Styles[0].Font.alignment:=3;
  Usrlr.Styles[2].font.color:=clblue;

  u:=UsrLr.NewObject;
  pt.x:= 0;
  pt.y:= -20;
  u.style:=0;
  begin
    U.Caption:='Sample1';
```

```
    U.x:=-80;
    U.y:=45;
    // Usrlr.styles[0].mark.bitmap.handle:=bm.handle;
    UsrLr.styles[0].Mark.bitmap.LoadImage('redbtn.bmp');
  end;

  u:=UsrLr.NewObject;
  pt.x:= 0;
  pt.y:= 0;
  u.style:=2;
  begin
    U.Caption:='Sample2';
    U.x:=-80;
    U.y:=47;
    // Usrlr.styles[2].mark.bitmap.handle:=bm.handle;
    UsrLr.styles[2].Mark.bitmap.LoadImage('redbtn.bmp');
  end;

  Panel2.font.color:=clblack;
  UsrLr.upper:=5000;
  UsrLr.Styles[0].Mark.upper:=5000;
  MapTivate1.redrawmap;
  MessageBeep(0);
  bm.free;
  Panel2.caption:='# Items[b] = '+inttostr(MapTivate1.UserDraw.count);
  with MapTivate1 do
  begin
    ZoomSysRect(UserDraw.Extents.xmin,UserDraw.Extents.ymin,
                UserDraw.Extents.xmax, UserDraw.Extents.ymax);
  end;
end;
```

StyleCount:Integer

The total number of currently defined styles in the UserDraw object. (Read-Only)

Upper:Double

Upper limit of visibility for the User Item theme.

Visible:Boolean

Defines if the UserDraw object is visible.

16. TSmObject Object

Interface for obtaining the the data corresponding to a given map object. In particular useful in the VB environment, where use of pointers is not supported to the extent they are supported in other environments. The interface to a TsmObject is obtained using the GetSmObject(n) method off the Themes interface.

Area:Double

The area enclosed by a polygon or polyline object. Similar to the GetArea, but this is a property of the map object.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject
  ' Display the area of the map object
  Nobj = Nobj + 1
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  Text1.Text = "Map Object#" & Str(Nobj) & ", Area (sq. mi)= " & Str(TempObject.Area)
End Sub
```

Centroid:TRPoint

The centroid of a polygon or polyline object. Similar to the GetCentroid, but this is a property of the map object.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject, TCent As TRPoint
  Nobj = Nobj + 1
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  'Mark the centroid of the object
  MapTivate1.UserPaint.SmPoint TempObject.Centroid.X, TempObject.Centroid.Y
  MapTivate1.RefreshMap
End Sub
```

Counts: array of Integer

Contains the count of points in each segment of a multi-segment object.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject
  Nobj = Nobj + 1
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  Text1.Text = "Map Object#" & Str(Nobj) & ", contains " & Str(TempObject.Num) & "
elements!"
End Sub
```

Data:String

A concatenated string with the contents of all the fields in the database, for the specified map object (includes delimeters).

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject
  ' Display the contents of ALL fields in the database
  Nobj = Nobj + 1
```

```
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  Text1.Text = "Object #" & Str(Nobj) & ", Field #1 " & TempObject.Data
End Sub
```

Fields:Array of string

An array containing the field definitions of the database the specified object belongs to.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject
  ' Display the contents of the first field in the database (country name)
  Nobj = Nobj + 1
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  Text1.Text = "Map Object#" & Str(Nobj) & ", Field #1 " & TempObject.Fields(1)
End Sub
```

GetClosestPoint (tp:LongInt; RefPoint:TRPoint; Dist:double):TRPoint

Returns the closest point to the specified object.

tp The type of calculation the user is interested in. The result depends on **tp** and the type of object being used in the calculation, as shown below:

	Type of Object		
Tp	0 – Point	1-Polyline	2-Polygon
0	Return Closest Vertex	Return closest interpolated point	If specified point is outside the object, return point on the perimeter, if inside, return specified point.
1	Return Closest Vertex	Return Closest Vertex	Return Closest Vertex
2	Return closest interpolated point (Treat object as polyline)	Return closest interpolated point (Treat object as polyline)	Return closest interpolated point (Treat object as polyline)
3	Treat object as polyline. If specified point is outside the object, return point on the perimeter, if inside, return specified point.	Treat object as polyline. If specified point is outside the object, return point on the perimeter, if inside, return specified point.	Treat object as polyline. If specified point is outside the object, return point on the perimeter, if inside, return specified point.

RefPoint The point from which the closest point on the object is obtained.

Dist The calculated distance to the closest point.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject
  Dim ClPt As TRPoint, ref0 As TRPoint
  ' Find closest point and draw line from 0,0 to that point
  Nobj = Nobj + 1
  ref0.X = 0
  ref0.Y = 0
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  ClPt = TempObject.GetClosestPoint(0, ref0, 100)
  MapTivate1.UserPaint.SmPoint ClPt.X, ClPt.Y
  MapTivate1.UserPaint.SmLine 0, 0, ClPt.X, ClPt.Y
  MapTivate1.RefreshMap
End Sub
```

GetPoint(Fraction:Double):TRPoint

Returns the point at a distance of fraction length along the object, e.g. if fraction=0.5 it returns the midpoint. If the specified object is a polygon, then the calculated fraction and point are of the "perimeter" of the object. Note, however, that if the object is composed of multiple polygons, the returned point is meaningless, since all the object perimeters are used in the calculation.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim T As TSmObject
  Dim ClPt As TRPoint, SomePoint As TRPoint
  ' Mark some points along the perimeter
  Nobj = Nobj + 1
  Set T = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  SomePoint = T.GetPoint(0.01)
  MapTivate1.UserPaint.SmPoint SomePoint.X, SomePoint.Y
  SomePoint = T.GetPoint(0.1)
  MapTivate1.UserPaint.SmPoint SomePoint.X, SomePoint.Y
  SomePoint = T.GetPoint(0.15)
  MapTivate1.UserPaint.SmPoint SomePoint.X, SomePoint.Y
  SomePoint = T.GetPoint(0.2)
  MapTivate1.UserPaint.SmPoint SomePoint.X, SomePoint.Y
  MapTivate1.RefreshMap
End Sub
```

ObjectIndex:LongInt

The Index # of the Object SMObject is currently pointing to.

Num

The count of Counts, i.e., the numner of elements in the Counts array. Also the number of segments/objects in a multi-object map object.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject
```

```
  Nobj = Nobj + 1
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  Text1.Text = "Map Object#" & Str(Nobj) & ", contains " & Str(TempObject.Num) & "
elements!"
End Sub
```

Perimeter:Double

The perimeter of a polygon or polyline object. Similar to the GetPerimeter, but this is a property of the map object.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject
  ' Display the perimeter of the map object
  Nobj = Nobj + 1
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  Text1.Text = "Map Object#" & Str(Nobj) & ", Perimeter (mi)= " &
Str(TempObject.Perimeter)
End Sub
```

Points:Array of TRPoint

An array containing all the points of the map object.

Sample Code (Visual Basic)

```
Private Sub Command28_Click()
  Dim TempObject As TSmObject
  ' Display the points of the object
  Nobj = Nobj + 1
  Set TempObject = MapTivate1.Themes("World-Countries").GetSmObject(Nobj)
  jj = -1
  For i = 0 To TempObject.Num - 1
    For j = 0 To TempObject.Counts(i) - 1
       jj = jj + 1
       List3.AddItem Str(TempObject.Points(jj).X) & ", " & Str(TempObject.Points(jj).Y)
    Next j
  Next i
  Text1.Text = "Map Object#" & Str(Nobj) & ", Perimeter (mi)= " &
Str(TempObject.Perimeter)
End Sub
```

ThemeIndex:LongInt

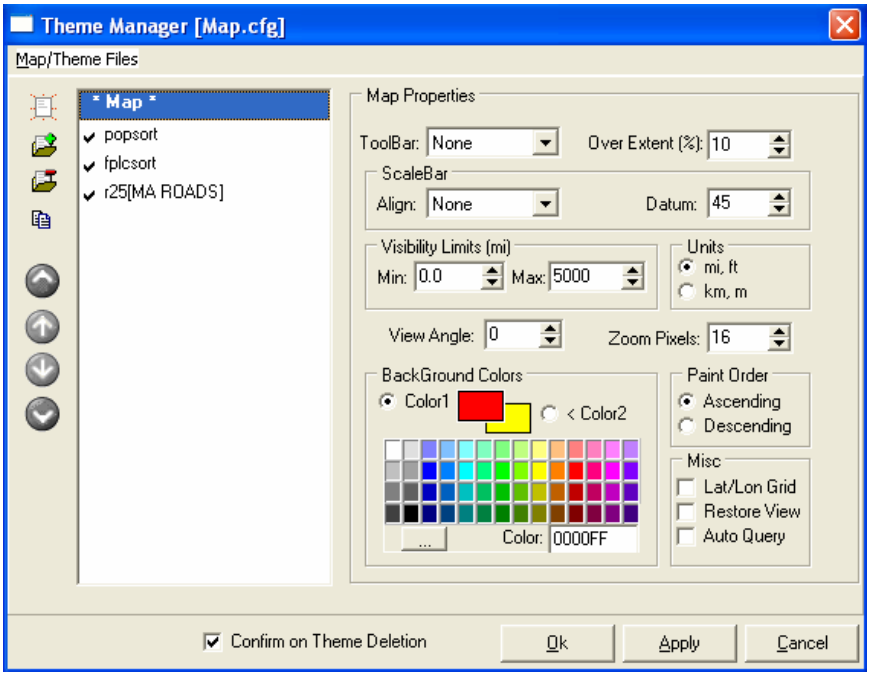
The Index number of the Theme that the object pointed to by SMObject, belongs to. This index number is the theme's order in the list of themes (Themes manager dialog).

Tp:LongInt

Type of map object. (1-Point, 2-Polyline, 3-Polygon)

17. ThemeManager Dialog

When the main dialog page opens, the name of the map (Theme collection) is highlighted and a number of options may be set that affect the overall map characteristics.



Most of the options in this page, in particular in the right portion of the file, are self explanatory.

Selecting **File/Theme Files**, opens a powerful menu list that permits the user to select the the following operations:

New Map Profile	Erases **all** information associated with the current Map definition, so the user can create a new map profile, with a new collection of themes.
Load Map Profile (.CFG)	Loads a complete Map definition profile from a .CFG file, replacing the one currently loaded/defined.
Save Map Profile (.CFG)	Saves the current map definition to the currently named Map profile (.CFG) file. If no name has been assigned, the user is prompted to enter a name.
Save Map Profile As (.CFG)	Saves the current map definition to a new Map profile (.CFG) file.
Load Theme (.TRT)	Load a theme definition (.TRT file). This file contains all the information defining a theme, including visibility thresholds, data files associated with it, etc. as well as all style definition information (which is also included in any saved style .TRU files). The theme is added to the list of currently defined

	themes.
Save Theme (.TRT)	Save the definition of the currently highlighted Theme to a theme definition (.TRT) file.
Exit	Close the drop down menu .

If the user right-clicks on a highlighted theme, then a pop-up menu appears with the following options:

Clear Map	Erases **all** information associated with the current Map definition, so the user can create a new map profile, with a new collection of themes.
Add Theme	Add a new theme to the theme set. A theme with the name "undefined" is added to the list and the second (theme setting) dialog page opens, to allow the user to set the properties of the new theme.
Delete Theme	Delete the currently highlighted theme from the theme set. Confirmation is required if the "Confirm on Theme Deletion" option is checked.
Clone Theme	Clone, i.e., create a duplicate of the highlighted theme with all its settings.
Modify Styles	The Styles dialog is opened, as if the "Modify Styles" button was pressed.
Apply Styles	Opens a standard File Open dialog and allows the user to load a Style (.TRS) file to be applied to the current theme.
Send to Top	Place the current Theme at the top of the list (painted first, if the default paint order is active).
Bring to Bottom	Place the current Theme at the top of the list (painted first, if the default paint order is active).
Move Up One	Move the current Theme up one position.
Move Down One	Move the current Theme down one position.

And, here is the functionality of the buttons:

Clear all current settings and start a new theme set (configuration file)

Add a new theme to the theme set. A theme with the name "undefined" is added to the list and the second (theme setting) dialog page opens, to allow the user to set the properties of the new theme.

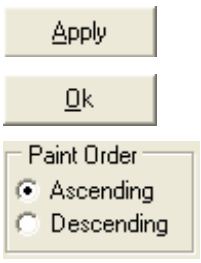
Delete the currently highlighted theme from the theme set. Confirmation is required if the "Confirm on Theme Deletion" option is checked.

Clone, i.e., create a duplicate of the highlighted theme with all its settings.

Move the currently highlighted Theme one position Up. (Active when a theme is highlighted)

Move the currently highlighted Theme one position Down. (Active when a theme is highlighted)

If checked, the user is asked to confirm any theme deletion from the map collection.

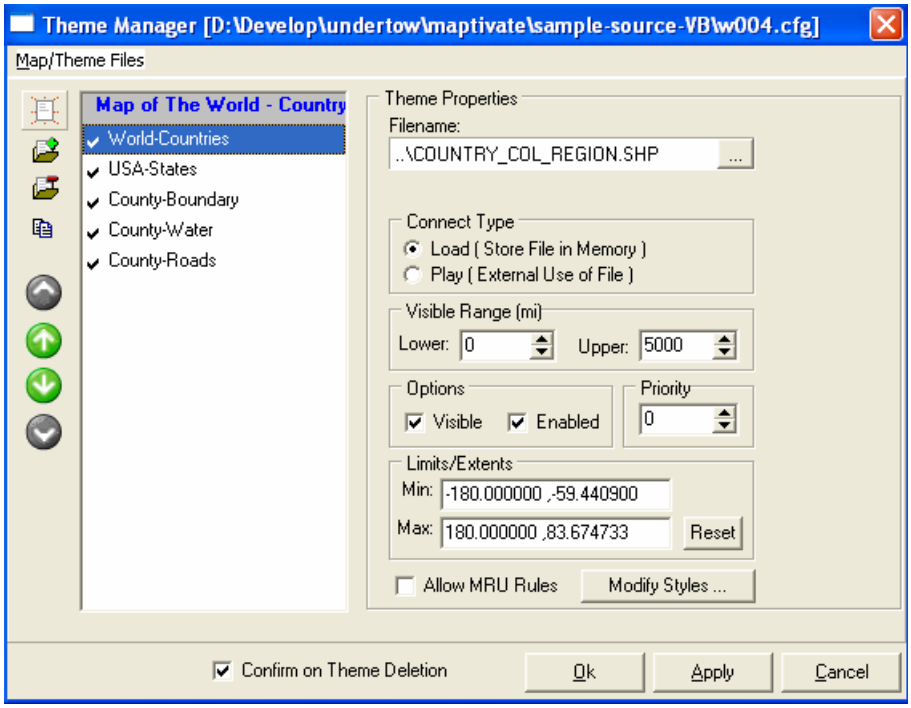


Apply any of the selection modifications made by the user, update the map, but leave the dialog open.

Apply any of the selection modifications made by the user, close the dialog and update the map.

Controls the order ot painting the themes in the map collection.

If one of the themes is highlighted, then the page that appears, on the right portion of the dialog, lets the user set options specific to "that" theme.

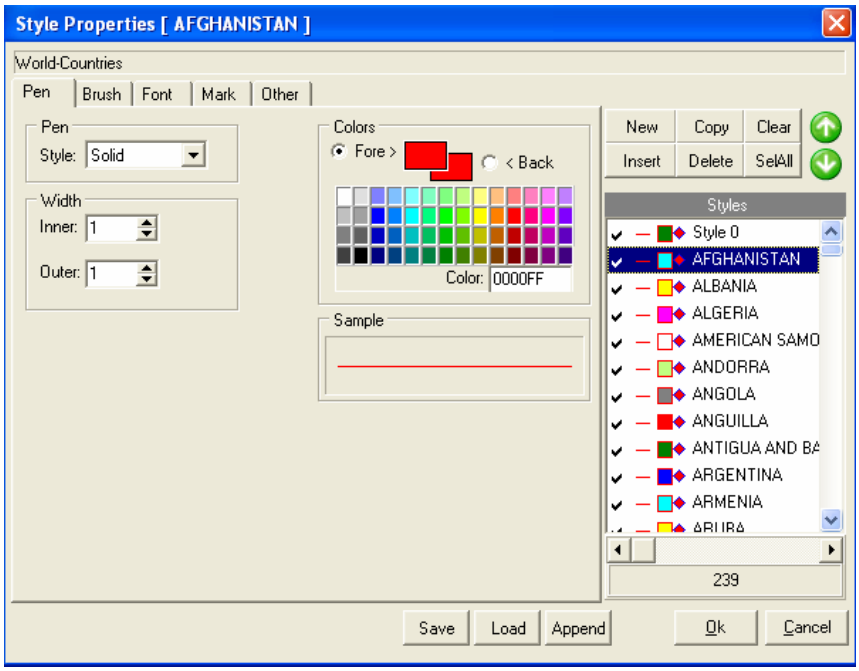


The user may navigate to the file to connect to this theme, decide whether to load it or play it, and set the visibility and scale parameters. The dialog also allows the user to modify the extents of the currently selected theme. This modification is temporary, and clicking the Reset button will restore the original extents stored in the data set.

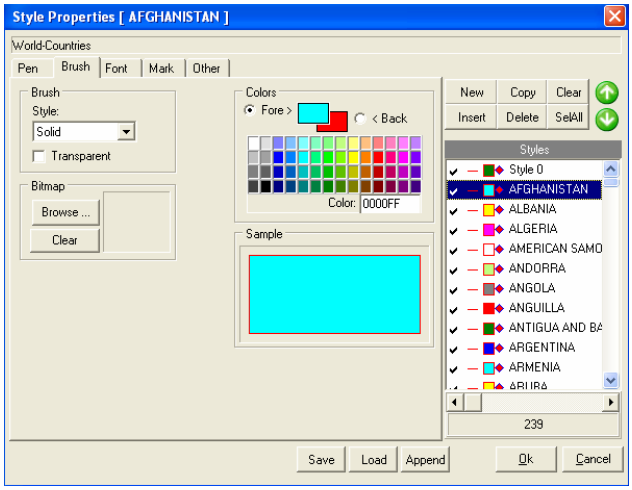
Double-clicking on a Theme name allows the user to edit that name in-place (the original name is automatically defined the same as the filename the theme is connected to – without the extension).

Clicking on the **Modify Styles** button, opens up the Styles dialog that allows the user to customize the look of the current theme. This dialog has a number of tabs, each allowing the user to modify a specific set of this theme's characteristics or (style). Any modifications made to these properties/characteristics, apply to all Styles selected on the right hand side of the dialog. The buttons **Save** and **Load**, common to all tabs in this dialog, allow the user to Save or Load a Styles (.TRS) file. See appendix "E" for a sample listing of such a file.

The first tab, in the Styles dialog, is the Pen attributes tab.

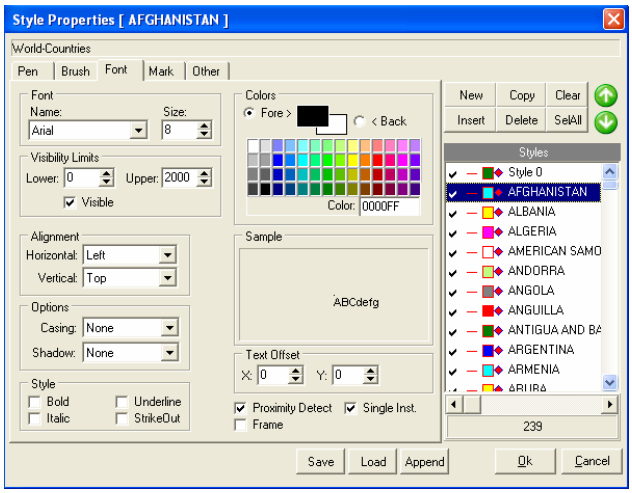


By specifying an outer line width larger than an inner one, the user may create the effect of a double-lined road. Clicking on the "Details" button, the user may hide/show the list of layers defined within the current style. DOUBLE Click on a Style to change Name. The **New, Copy, Delete** and Clear buttons allow the user to manage the Styles in a given theme.



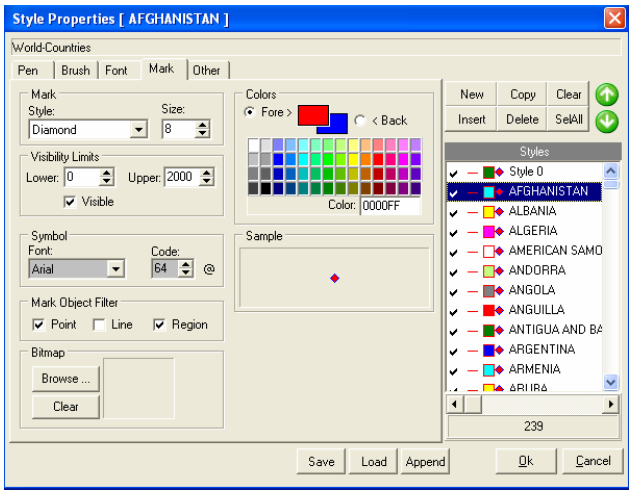
The user may select one of the predefined brush styles, or elect to use a custom defined shade pattern, specified through a user bitmap.

All modifications made to these attributes apply to all selected Styles. (Multiple styles may be selected using the standard Ctrl-Click, Shift-Click Windows methods.



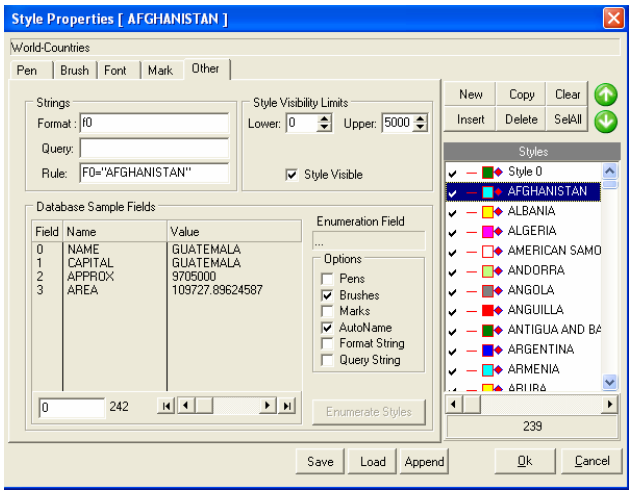
This tab controls all the properties of the font that is used to label objects in this theme.

Note its visibility thresholds and the visible flag. Also, note the Proximity Detect flag that employs Bounding Box Collision Detection (BBCD) to eliminate labels placed on top of other labels.



This tab controls the properties of the marker that used to identify Points, Lines and Regions. The user may select one of the pre-defined markers, or select any symbol from an already installed TTF.

Note that for Lines and Regions, the location of the marker is also the location of the text label.



Special Tab, allows the user to specify the Rules to be used for labeling, querying and assigning objects to a specific style.

Format is the specification of the FormatStr Poperty, **Query** the specification of QueryStr and **Rule** the specification of RuleStr. See **Theme Object** description for details.

This dialog also permits the user to see the fields, and scan through the records of the database associated with the currently selected Theme. The **Enumerate Styles** button is a powerful tool for thematically

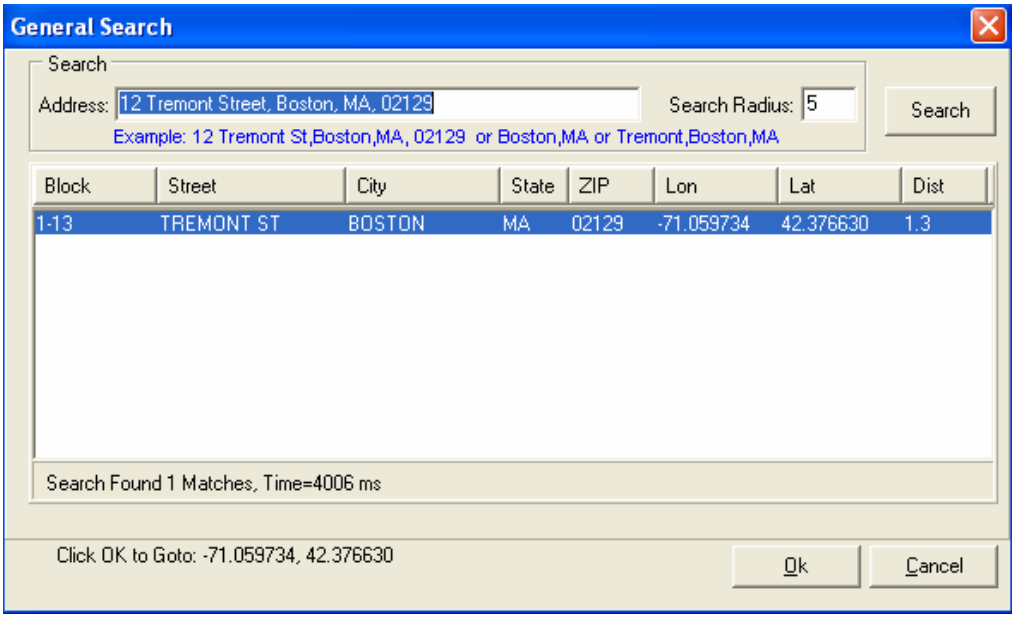
assigning attributes. It scans the Highlighted Database field, and finds all unique entries. It then creates a Style for each such entry and populates the Styles list with them. If **the Pens, Brushes and Marks** options are checked, it also cycles through 8 colors and assigns them to each generated Style. If the AutoName option is selected, then it replaces the default generated Style name (Style0, Style1,…) with the actual enumerated value that is being used for the style.

Rule-based styles are evaluated at Run time and are based on the Rule String. Rules are traversed from lowest to highest, through all Visible styles. The first rule that results in a non-zero outcome will be used as the style for that object.

In the example shown in the figure above, for instance, the Label "+F0" indicates that the elements of the theme (countries) would be labeled with the 0-th field (country) name. The RuleStr string, F0="ANGOLA" signifies that the highlighted rule attributes (Angola) will be applied to an object that satisfies the rule, i.e., the value of its zeroth data field is "ANGOLA".

18. SearchDlg Dialog

A highly specialized dialog that is used for performing address/place searches when using the USA, Tiger-based data set distributed by Undertow Software. The data structure in the files distributed as part of the above data set, is a prerequisite for the search tied to this dialog to work as intended.



There are only two user input fields. The Search Address and the Search Radius. The Search address can be any valid address, composed by one or more of the following components, separated by a comma:

Address #, Street Name, City, State Abbreviation, ZipCode

MapTivate attempts to determine what the user is searching for, before it passes the search string to the search algorithms.

The following are examples of valid/typical search Strings:

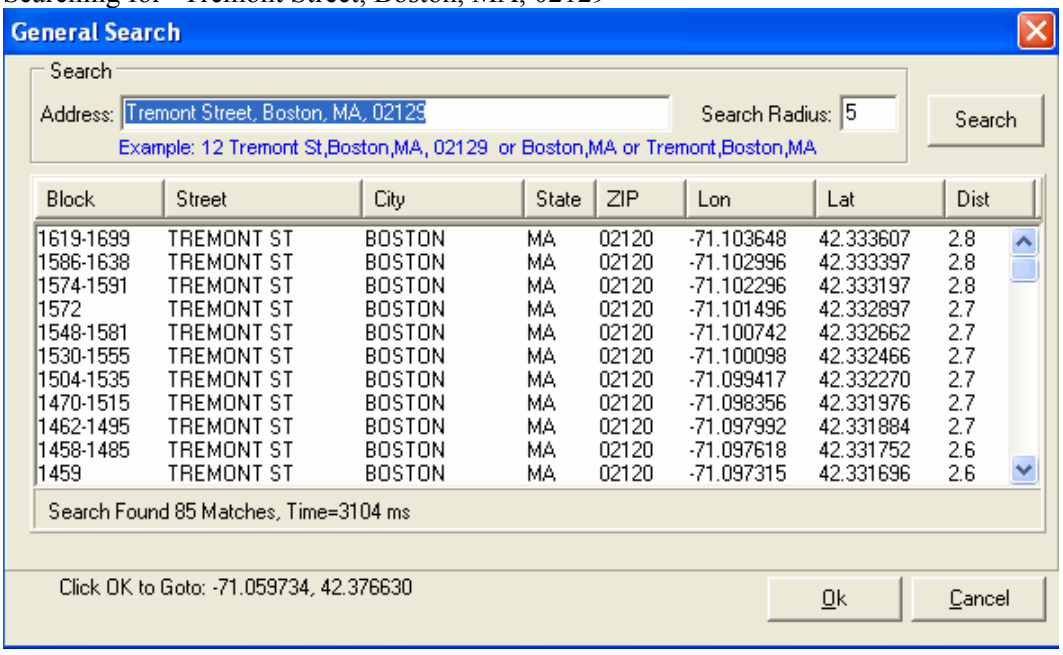
12 Tremont Street, Boston, MA, 02129
Tremont Street, Boston,MA
Boston
Market St., Lowell, MA
Market Street, Lowell, MA, 01852
elizabeth way, lynnfield,ma
thwing rd,lynnfield,ma
Thwing Road,Lynnfield,MA,01940
hynes lane,maynard,ma
Hynes Ln,Maynard,MA
Springfield
Chicago,IL
Chicago

The Search radius specifies how far from the point identified by the first level search, the OCX should continue to look in trying to narrow down hits meeting the specified criteria. For example, if the user entered "Market St., Lowell, MA", and a search radius of 2.0, then the OCX will first find the coordinates of the point corresponding to "Lowell, MA", and will then search and retur all hits of Market Street within 2.0 miles from the isedntified point.

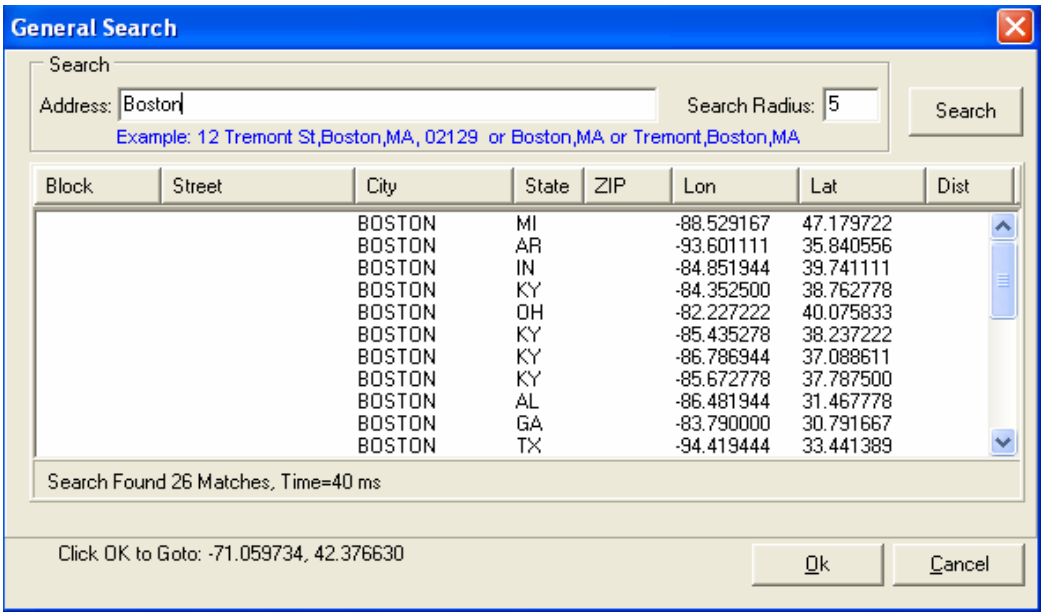
Adding a ZipCode makes restricts the search area from the ZipCode centroid, and r eusually results in faster searches.

The results of a couple of such searches are shown in the captured images of the Search dialog shown below:

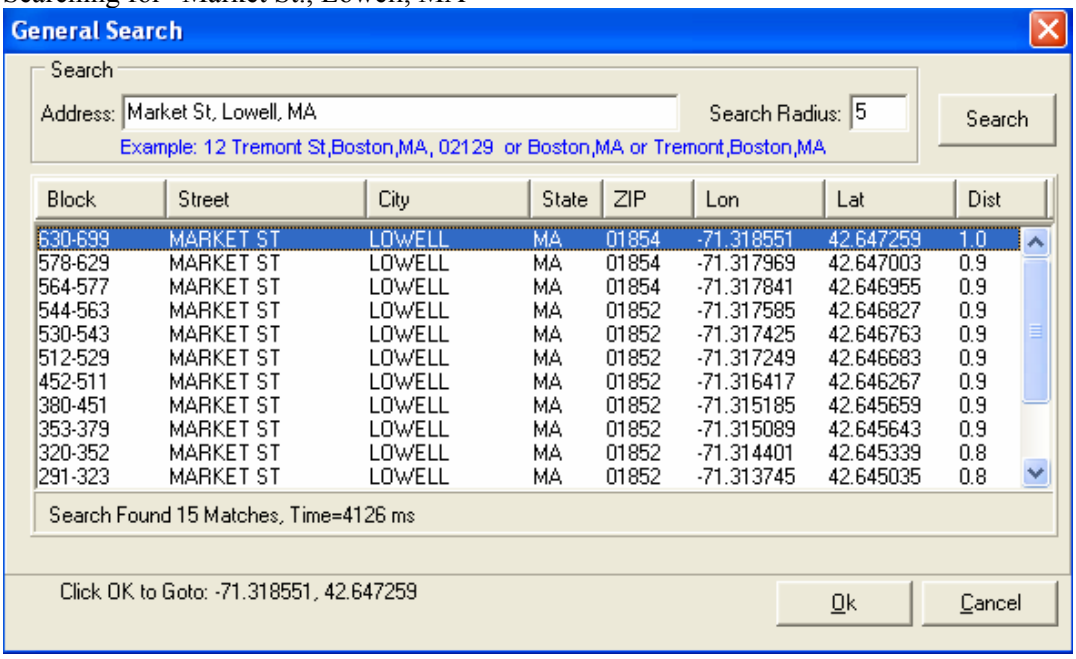
Searching for "Tremont Street, Boston, MA, 02129"



Searching for ***"Boston"***



Searching for "Market St., Lowell, MA"



Clicking on one of the headings (Street, City, etc.) sorts the outout results using that field as the key.

Double clicking on one of the returned hits, places it in the lower portion of the dialog, and when OK is pressed, and the dialog closes, the viewport is re-centered about its coordinates. Hitting Cancel, instead of OK, simply closes the dialog.

19. TRBitmap Object

Defines the properties of the bitmap to be used. Note that 8, 16 and 24-bit windows (BMP) bitmaps can be used. The lower left corner pixel color is assumed to be the transparent color.

Angle:Integer

Set the angle (degrees) that the bitmap should be rotated. Positive sense is counterclockwise.

Sample Code (Delphi 5)

```
procedure TForm1.UserDrawObjClick(Sender: TObject);
var n,m:integer;
    bm:tbitmap;
    d:trect;
    pt:trpoint;
    X,y:double;

Begin
  // Two Styles are added to be used later
  UsrLr.AddStyle;
  UsrLr.Addstyle;
  UsrLr:=MapTivate1.UserDraw;
  USrLr.Styles[0].Font.color:=clred;
  UsrLr.Styles[0].Font.alignment:=3;
  Usrlr.Styles[2].font.color:=clblue;

  u:=UsrLr.NewObject;
  u.style:=0;
  begin
    U.Caption:='Sample1';
    U.x:=-80;
    U.y:=45;
    UsrLr.styles[0].Mark.bitmap.LoadImage('bin.bmp');
    UsrLr.styles[0].Mark.bitmap.angle:=10;
  end;
  u:=UsrLr.NewObject;
  u.style:=2;
  begin
    U.Caption:='Sample2';
    U.x:=-80;
    U.y:=47;
    UsrLr.styles[2].Mark.bitmap.LoadImage('bin.bmp');
    UsrLr.styles[2].Mark.bitmap.angle:=5;
  end;
  MapTivate1.ZoomCenter(-80,46);
  Panel2.font.color:=clblack;
  UsrLr.upper:=5000;
  UsrLr.Styles[0].Mark.upper:=5000;
  MapTivate1.redrawmap;
  MessageBeep(0);
  bm.free;
  Panel2.caption:='# Items[b] = '+inttostr(MapTivate1.UserDraw.count);
end;
```

Handle:Integer

For using bitmaps already loaded, rather than loading a new instance from a bitmap file. See sample code for ***.Angle***, above.

Height:Integer

The height of the bitmap in pixels. (Read Only)

LoadImage(FileName:String)

Loads a bitmap (.BMP, or .GIF) file to be used by the control.

Sample Code (Delphi 5)

```
procedure TForm1.SetMarksClick(Sender: TObject);
begin
with MapTivate1.Themes[0] do
  begin
    styles[0].brush.style:=1;
    // Setting Mark Attributes
    with styles[0].mark do
    begin
      VisFilter:=4;
      Visible:=true;
      Style:=-1;
      bitmap.LoadImage('redbtn.bmp');
      Upper:=20;
      Lower:=0.1;
      Color:=clBlue+IMrkStyle*100;
      BorderColor:=clRed;
      Size:=10+IMrkStyle*4;
    end;
  end;
  MapTivate1.RedrawMap;
end;
```

Transparent:Boolean

Determines whether the bitmap will be considered as transparent, or not. If set to True, then the color of the lower left corner pixel of the bitmap is considered the transparent color.

Visible:Boolean

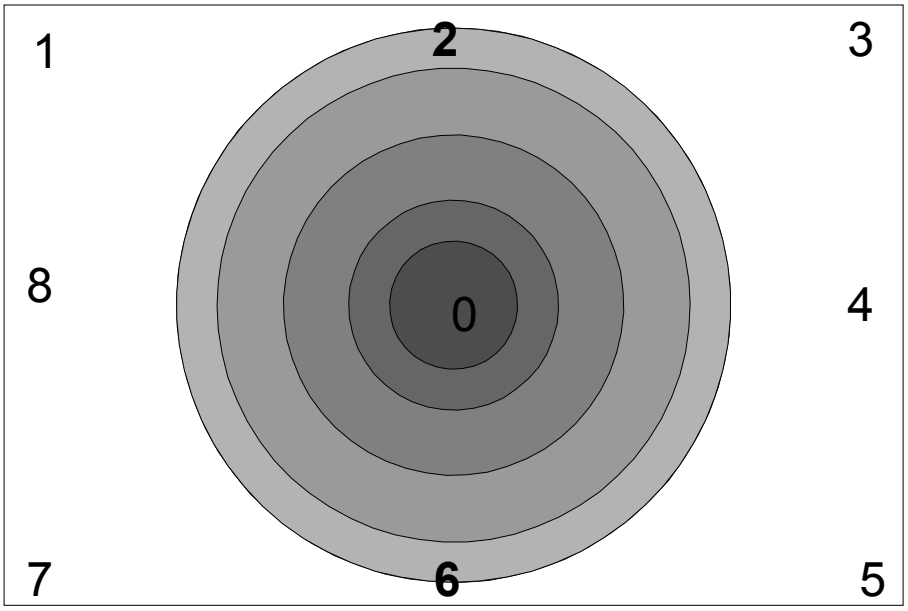
Sets the visibility of the bitmap.

Width:Integer

The width of the bitmap in pixels (Read Only)

20. BackGround Object

The map control may be painted with a background pattern specified by the user. This pattern is defined by a center point color that transitions to a Dither color moving away from a user-specified reference point.



Moving Away from CenterPoint

ColorCenter:Integer

Starting color to create the dithering pattern.

ColorDither:Integer

Color to be mixed with the Center (starting) color and predominates as the pattern is moving away from the reference point.

Align:Integer

Alignment position for the starting point of the dither pattern. An alignment position 0 to 8 may be specified, as shown in the diagram at the beginning of this section.

Sample Code (Delphi 5)

```
procedure TForm1.SetColsClick(Sender: TObject);
```

```
begin
  inc(aLmode);
  MapTivate1.Background.ColorCenter:=clblue;
  MapTivate1.BackGround.ColorDither:=clyellow;
  MapTivate1.BackGround.Align:=alMode;
  MapTivate1.BackGround.Decay:=2;
  panel7.caption:='Background Align:'+inttostr(alMode);
  MapTivate1.RedrawMap;
  MapTivate1.BorderColor:=clgreen;
end;
```

Decay

Speed by which the dithering process goes from the CenterColor to the DitherColor. Zero will result in the ColorCenter filling the whole control surface, while a very large value results in the Dither color filling the whole control surface.

Sample Code (Visual Basic)

```
Private Sub Command38_Click()
  BkgSet = Not (BkgSet)
  If BkgSet = False Then
    MyBkAlign = MapTivate1.BackGround.Align
    MyBkColorCenter = MapTivate1.BackGround.ColorCenter
    MyBkCOlorDither = MapTivate1.BackGround.ColorDither
    MyBkDecay = MapTivate1.BackGround.Decay
    MapTivate1.BackGround.Align = bkCenter
    MapTivate1.BackGround.ColorCenter = vbBlue
    MapTivate1.BackGround.ColorDither = vbYellow
    MapTivate1.BackGround.Decay = 20
    MapTivate1.RedrawMap
  Else
    MapTivate1.BackGround.Align = MyBkAlign
    MapTivate1.BackGround.ColorCenter = MyBkColorCenter
    MapTivate1.BackGround.ColorDither = MyBkCOlorDither
    MapTivate1.BackGround.Decay = MyBkDecay
    MapTivate1.RedrawMap
  End If
  Beep
End Sub
```

21. TRViewMgr Object

Allows the user to create and manipulate multiple view settings, and then quiccky restore these views.

Add(S:TviewRec);

Adds the user specified view to the views list object.

Sample Code (Delphi 5)

```
procedure TForm1.AddViewClick(Sender: TObject);
var MyVu:TViewRec;
begin
  // Define a view
  MyVu.Name:='My First View';
```

```
  MyVu.xref:=-70.5;
  MyVu.yref:=42;
  MyVu.Zoomscale:=5;
  // Add View at the end of current Views list
  MapTivate1.ViewMgr.Add(MyVu);
  // Display dialog to see New View added
  MapTivate1.ViewMgr.Dialog;
end;
```

CaptureView(S:String)

Captures the current view and adds it to the views list object.

Sample Code (Delphi 5)

```
procedure TForm1.CaptureViewClick(Sender: TObject);
begin
  // Capture Current View
  MapTivate1.ViewMgr.CaptureView('InLine Captured View');
end;
```

Clear()

Clear all Views and release all resources.

Sample Code (Delphi 5)

```
procedure TForm1.AddViewsClick(Sender: TObject);
var MyView01, MyView02 :TViewRec;
begin
  // Clear all current Views from List object
  MapTivate1.ViewMgr.clear;
  // Define First View
  MyView01.Name:='My First View';
  MyView01.xref:=-70.5;
  MyView01.yref:=42;
  MyView01.miles:=5;
  // Add View at the end of current Views list
  MapTivate1.ViewMgr.Add(MyView01);
    // Define Second View
  MyView02.Name:='My Second View';
  MyView02.xref:=-70.5;
  MyView02.yref:=42;
  MyView02.miles:=5;
  // Add View at the end of current Views list
  MapTivate1.ViewMgr.Add(MyView02);
  // Display dialog to see New Views added
  MapTivate1.ViewMgr.Dialog;
end;
```

Count

Number of Views in the Views object.

Sample Code (Delphi 5)

```
procedure TForm1.AddViewsCritClick(Sender: TObject);
var MyView :TViewRec;
begin
  // Check to see ho many views are already in the list.
```

```
  // If the are less than 5, then add a new one
  If MapTivate1.ViewMgr.Count<5 then
  begin
  // Define First View
    MyView.Name:='My First View';
    MyView.xref:=-70.5;
    MyView.yref:=42;
    MyView.miles:=5;
    // Add View at the end of current Views list
    MapTivate1.ViewMgr.Add(MyView);
  end;
  // Display dialog to see New Views added
  MapTivate1.ViewMgr.Dialog;
end;
```

Delete(N:Integer)

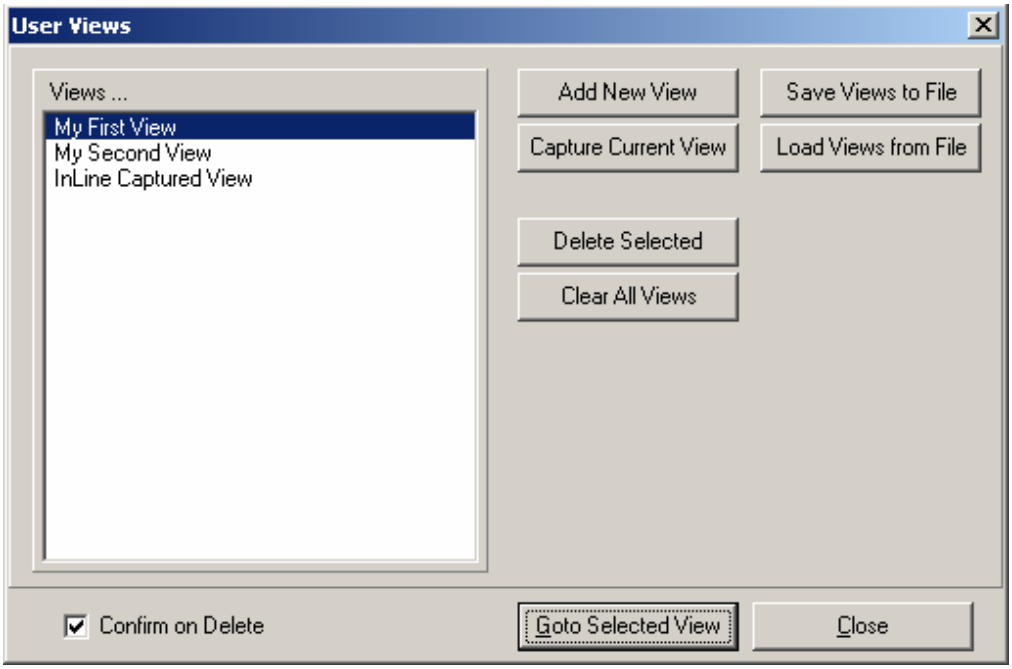
Delete specified N-th view from views list.

Sample Code (Delphi 5)

```
procedure TForm1.DeleteOldViewClick(Sender: TObject);
begin
  // Delete the second view from the list.
  MapTivate1.ViewMgr.Delete(1);
end;
```

Dialog()

Opens the Views dialog.



Items[Index:integer]: TViewRec

Indexed List of all defined views.

Sample Code (Delphi 5)

```
procedure TForm1.SetSecViewClick(Sender: TObject);
begin
  with MapTivate1.ViewMgr do
  begin
     //  Go to the second view (0-based)
    SetViewByName(items[1].Name);
  end;
end;
```

LoadFromFile(S:String)

Loads the views definitions from the file specified by the user. If no file extension is specified, then the extension .TVW is used.

Sample Code (Delphi 5)

```
procedure TForm1.LoadViewFileClick(Sender: TObject);
begin
  // Load a views file
  MapTivate1.ViewMgr.LoadFromFile('MyTempViews');
end;
```

SaveToFile(S:String)

Saves the current views definitions to the file specified by the user. If no file extension is specified, then the extension .TVW is used.

Sample Code (Delphi 5)

```
procedure TForm1.SaveViewFileClick(Sender: TObject);
begin
   // Save currently defined views to a file
   MapTivate1.ViewMgr.SaveToFile('MyTempViews');
end;
```

SetViewByName(S:String)

Sets the Viewport parameters corresponding to the View specified by **S**.

Sample Code (Delphi 5)

```
procedure TForm1.SetSecViewClick(Sender: TObject);
begin
  with MapTivate1.ViewMgr do
  begin
     //  Go to the second view (0-based)
    SetViewByName(items[1].Name);
  end;
end;
```

22. MakeCRA – Archiver Tool

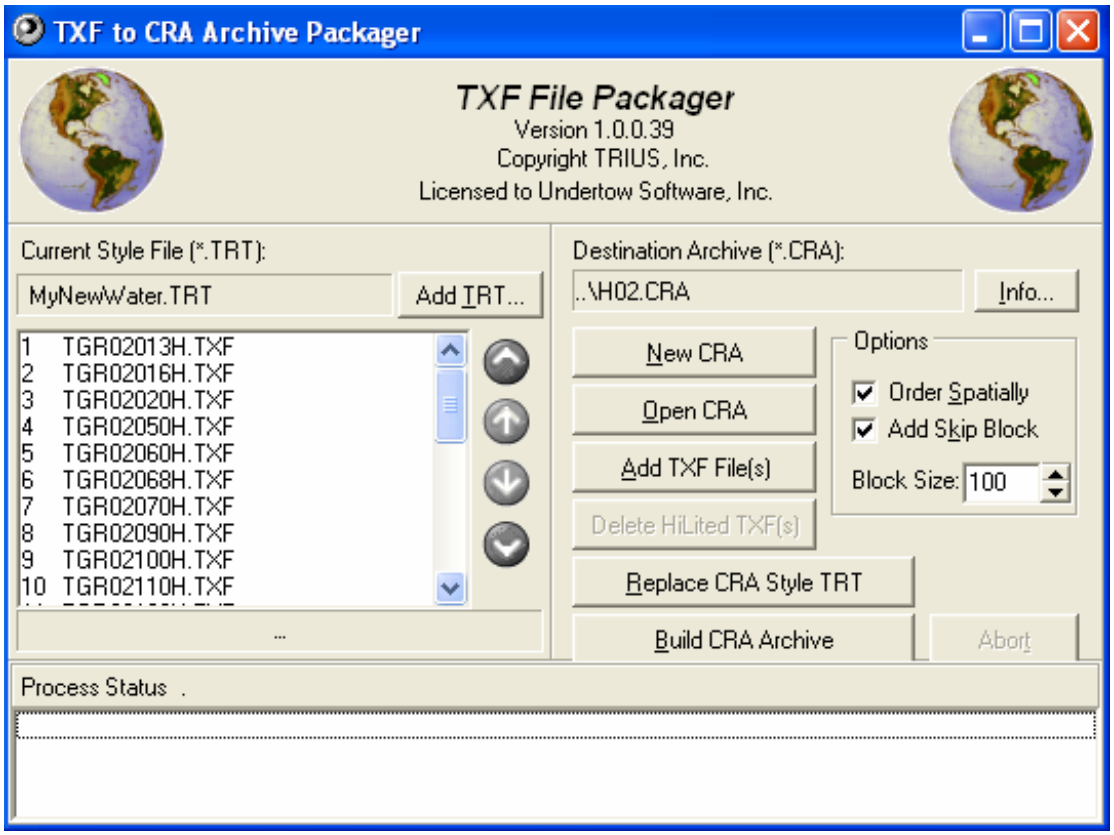
MakeCRA is a utility program used to produce CRA-type archive files used by the MapTivate Mapping/GIS tool from available from Undertow Software, Inc, using proprietary TXF/TDB* format data file pairs.

23. MakeCRA Operation

When the program is started, the user is presented with the simple dialog. This is sort of a wizard-dialog, in that it attempts to control the sequence of actions of the user, in order to minimize errors. The sequence for creating a CRA file is :

Start/Open a new CRA file
Add/Remove TXF/TDB files To/From it
Import a TRT (Theme) file, if desired
Specify the desired options
Build the CRA Archive
Get a brief view of the archive contents, if desired

The wizard-dialog used by the program is shown below and its various options are described for the user. It should be noted that when the program is first started it looks for a file CRAPack.cfg which contains last session's configuration options. The file is also automatically written out to the current default directory when the program is exited.



Description of Dialog Options

New CRA

Opens a standard Windows "File Open" dialog and permits the user to specify a new CRA file to create. If no CRA file is currently specified, all non-pertinent button options are disabled. Note that when the program is first loaded, it automatically loads its configuration file (CRAPack.cfg), from the current directory. If such file is detected and loaded, a CRA file may already be specified and its name is displayed in the "Current CRA" area.

Open CRA

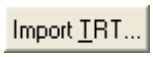
Opens a standard Windows "File Open" dialog and permits the user to open an existing CRA file. Once loaded, the contents of the CRA file may be views (see Info button below), or a new theme (.TRT) file may be imported into it. If no CRA file is currently specified, all non-pertinent button options are disabled. Note that when the program is first loaded, it automatically loads its configuration file (CRAPack.cfg), from the current directory. If such file is detected and loaded, a CRA file may already be specified and its name is displayed in the "Current CRA" area.

Add TXF File(s)

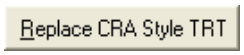
This button is disabled, unless a CRA file has already been specified/loaded. It opens up a standard Windows File Open dialog and permits the user to select TXF files to be added to the CRA archive. Note that if a TXF file is specified and its accompanying TDB file is not detected ,

an appropriate error message is displayed.

Delete highlighted TXF files from the CRA collection list. It does NOT affect the original TXF source files. Also, note that for any files to be deleted from the CRA, the CRA Build Operation needs to take place.

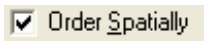


Opens up a standard Windows File Open dialog and allows the user to specify a TRT (Theme) file to be imported and added to the CRA file. Note that the actual TRT files are created and modified using the MapTivate control. Also note that the CRA file needs to be recreated (using the Replace CRA Style TRT button), after a new TRT file is imported into it.



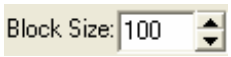
Replaces the TRT file with the currently inserted one, and re-creates the CRA file, without re-compiling all the TXF/TDB files

Build/Create the specified CRA file using the TXF, TRT files and the options specified by the user. The steps of the building process are echoed in the listbot at the bottom of the dialog.



Orders tha data in the archive spatially. In most cases, this is preferred, as it can significantly increase rendering speed, since objects in the current viewport, are near each other, thus minimizing the overhead of seeking to various portions of the file(s) to retrieve them.

A Skip-Block approach is embedded in the header of the created file, that allows the MapTivate tool to quickly position into the component TXF files, thus increasing rendering speed.



The number of objects in each skip block. A typical value of 100 is reasonable for most cases. However, the user may experiment with different block sizes for specific data sets.

Normally disabled, this button allows the user to abort the operation of creating a CRA file.

Move highlighted TXF files to the top of the list.

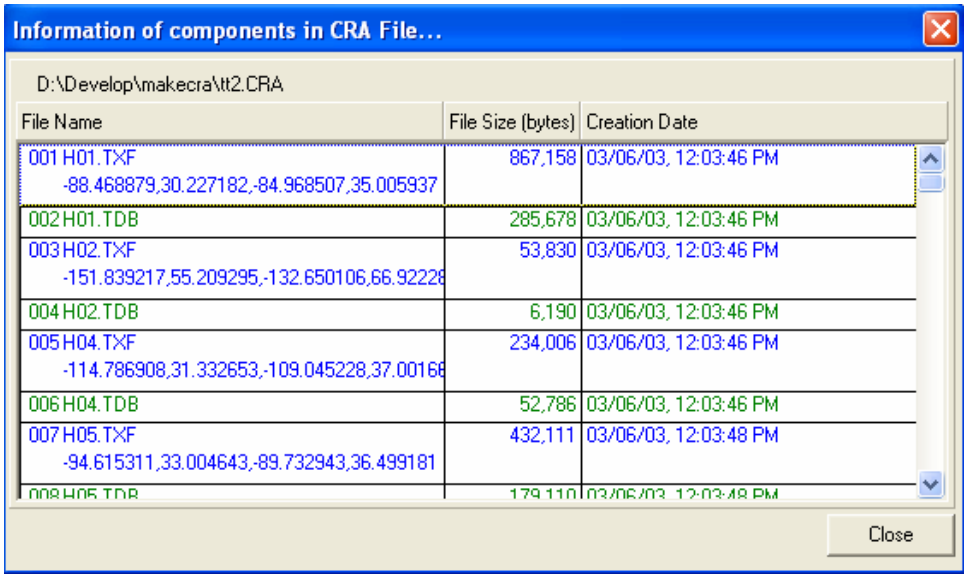
Move highlighted TXF files Up one position.

Move highlighted TXF files Down one position.

Move highlighted TXF files to the bottom of the list.

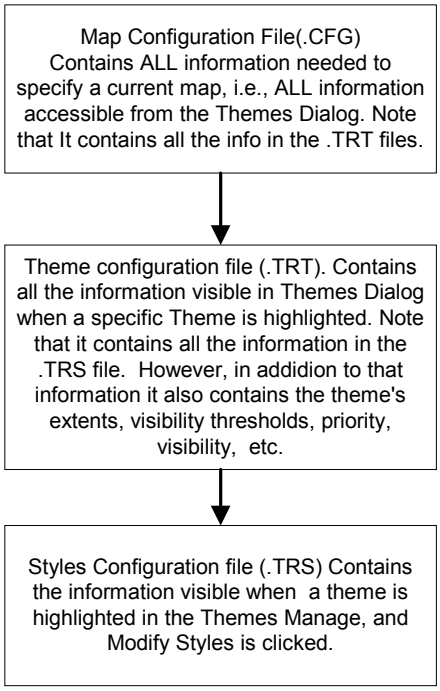
Info...

Open a form that displays summary information about the TXF files inside the current CRA Archive. It displays the name of the TXF and its accompanying TDB file, Their file size and date of creation, as well as the Lat/Lon extents of each TXF file. See below for sample screen capture.



24. Appendix “A” - Sample .CFG file listing

Before providing a listing of a sample .CFG file, it may be worthshile to re-iterate the relationship between the different types of control/configuration files saved by the MapTivate.



Sample listing of a .CFG file

```
MAP:My Sample Map 01
  SMAX:5000.0
  SMIN:0.0
  ANGLE:0.0
  TOOLBAR:0
  GRID:0
  ENABLED:1
COLOR:$8000000F
BCTR:$0000FF
BDTH:$FF0000
BCOLOR:$F0FEFF
  PORDER:0
  ALIGNBAR:1
  RESTVIEW:1
  OVEREXTEND:5
  CFMDEL:1
VIEW:-69.890823,41.420141,33.283153
LIMIT:-71.416337,40.195625,-68.365309,42.644657
EXTENTS:0.000000,0.000000,0.000000,0.000000
```

```
END
THEME:roads
  FILE:D:\Develop\smpmap\roads.mif
  NAME:roads
  EXTENTS:-70.686821,41.515479,-69.928591,42.081367
  CONNECT:0
  VISIBLE:1
  ENABLED:1
  UPPER:5000.0
  LOWER:0.0
STYLE=0
  NM=Style 0
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|15768
  BRUSH=0|008000|0000FF|925972017|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=
END
STYLE=1
  NM=Style 1
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|008000|1|0000FF|0|15184
  BRUSH=0|FF0000|0000FF|925972017|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|FFFF00|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A11"
END
STYLE=2
  NM=Style 2
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FF0000|1|0000FF|0|14328
  BRUSH=0|FFFF00|0000FF|16|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|00FFFF|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A15"
```

```
END
STYLE=3
  NM=Style 3
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FFFF00|1|0000FF|0|13504
  BRUSH=0|00FFFF|0000FF|49|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|FF00FF|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A17"
END
STYLE=4
  NM=Style 4
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|00FFFF|1|0000FF|0|12680
  BRUSH=0|FF00FF|0000FF|572664866|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|FFFFFF|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A21"
END
STYLE=5
  NM=Style 5
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FF00FF|1|0000FF|0|11856
  BRUSH=0|FFFFFF|0000FF|11768|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|000000|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A25"
END
STYLE=6
  NM=Style 6
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FFFFFF|1|0000FF|0|11032
  BRUSH=0|000000|0000FF|38919256|0
```

```
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|808080|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A31"
END
STYLE=7
  NM=Style 7
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|000000|1|0000FF|0|10208
  BRUSH=0|808080|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|000000|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A33"
END
STYLE=8
  NM=Style 8
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|808080|1|0000FF|0|9384
  BRUSH=0|000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|40140000|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A35"
END
STYLE=9
  NM=Style 9
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|8560
  BRUSH=0|40140000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|000000|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A41"
END
STYLE=10
  NM=Style 10
  VS=255
```

```
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|008000|1|0000FF|0|7736
  BRUSH=0|000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|40040000|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A42"
END
STYLE=11
  NM=Style 11
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FF0000|1|0000FF|0|6912
  BRUSH=0|40040000|0000FF|38928488|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|000000|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A43"
END
STYLE=12
  NM=Style 12
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FFFF00|1|0000FF|0|6088
  BRUSH=0|000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|40000000|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A44"
END
STYLE=13
  NM=Style 13
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|00FFFF|1|0000FF|0|5264
  BRUSH=0|40000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|000000|000000|1|128|2000.000000|0.000000
  FORMAT=
  QUERY=
```

```
  RULE=F4="A51"
END
STYLE=14
  NM=Style 14
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FF00FF|1|0000FF|0|4440
  BRUSH=0|000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|3FF00000|000000|1|76|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A52"
END
STYLE=15
  NM=Style 15
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FFFFFF|1|0000FF|0|11924
  BRUSH=0|3FF00000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|000000|000000|1|76|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A63"
END
STYLE=16
  NM=Style 16
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|000000|1|0000FF|0|11100
  BRUSH=0|000000|0000FF|65|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|3FF00000|000000|1|76|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A64"
END
STYLE=17
  NM=Style 17
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|808080|1|0000FF|0|10276
```

```
  BRUSH=0|3FF00000|0000FF|16|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|000000|000000|1|76|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A71"
END
STYLE=18
  NM=Style 18
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|9452
  BRUSH=0|000000|0000FF|16777217|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|40000000|000000|1|76|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="A74"
END
STYLE=19
  NM=Style 19
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|008000|1|0000FF|0|0
  BRUSH=0|40000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|000000|000000|1|0|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="F83"
END
STYLE=20
  NM=Style 20
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FF0000|1|0000FF|0|0
  BRUSH=0|000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|40040000|000000|1|0|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F4="F84"
END
END
SCALEMGR
```

END

25. Appendix “B” - Sample UserDraw file listing

```
STYLE=0
  NM=Style 0
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|0
  BRUSH=0|008000|0000FF|0|0
  FONT=Arial|8|0000FF|FFFFFF|0|3|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|0|5000.000000|0.000000
MKBMP=0|000000|000000|0
        AALz
        Fk1CAAACAAAAADYAKAAAAAAAAAANAAwAAQAAABgAAAAAAeAAAAAAAAAA
AAAAAAAA
        AAAAAAAA1tbW1tbW1tbW1tbW1tbW1tbWxsbGxsbGzs7O1tbW1tbW1tbW1tbW1tYA
        1tbW1tbW1tbWxsbWa2PGSkJaSkJCc2tCtbVrzs611tbO1tbW1gDW1tbW1tbWnNbW
        MZylOTk5QlI5UmNCUntSOXNSjEJCxoyU1sbGANbW1tbW1tbWzs7OOUIxhFpaa0JC
        SjExWjk5c1JShFpaOTkxpaWlzs7O1tYA3t7We3vee3t7hIScUlKtMTFzISFSKSk5
        SkpCWlprc3OExsZz1gDG79bWSu/vvUJSpda9Y8alQoxjMWNCOUIxMVI5WkoxUoRa
        tVJaALW13t7e7+/vQlJK3tbW1r29rYSEc0pKWjk5UjExQikpe1JSUlpStbW11tYA
        5+fWc2vnxs5rzs7OpaXeY2PGQkKMMTFjOTlKUlJSc3Nrvb1r1gC93tbW3t7ea97e
        3mtzvefejNa9WrWMQoRaQlpCMWNCnDE5zpycAM7O1tbW1tbW9/f3xsbGjIyM1s7W
        zq2trYSEc1paMTkxhISExsbG1tbW1tYA1tbW1tbW7+/Wxsbva2PGUkpaSkJCWlJC
        nJxSxsaczs7G1tbO1gDW1tbW1tbW1tbW3tbW597ezufnxs7OzsbGzs7O1s7O1tbW
        1tbWANbW
  FORMAT=
  QUERY=
  RULE=
END
STYLE=2
  NM=Style 2
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|6632
  BRUSH=0|008000|0000FF|102|0
  FONT=Arial|8|FF0000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|48|2000.000000|0.000000
MKBMP=14149936|D7E930|001780|538976288
```

AApH
Vk1CAAAHAAAAADYAKAAAAAAAAAAZABgAAQAAABgAAAAAByAAAAAAAAAA
AAAAAAAA
AAAAAAAA19fX19fX1tbW1tbW1tbW1tbW1tbW1NTU09PTz8/PysrKycnJysrK
y8vL0NDQ09PT1dXV1dXV1tbW1tbW1tbW1tbW1tbW19fX19cA1tbX1tbW1tbW1tbW
1tbW1tbW1tbW0NDWysrQw8PKvLzDubm8t7e5t7e3u7u3w8O7y8vD0tLL1NTS1NTU
1dXU1tbV1tbW1tbW1wDW1tfX1tbW1tbW1tbW1dbW1NXVxdTUkcXFZZGRS2VlREtL
RURERkVFT0ZGb09Pkm9vtZKSwbW1y8HB0cvL1NHR1dTU1tXV1tbWANbW1tbW1tbW
1tbW1tbW19fXz8/Pp6enXl5eRUVFQEBAOzs7OTk5Pz8/RUVFSUlJVFRUV1dXY2Nj
oqKiuLi4x8fH0NDQ1NTU1dXV1tbW1tYA1tbW1tbW2NjWy8vYj4/LYGCPNTVgOTk1
Q0M5SEhDS0tIVVVLX19VYmJfXl5iTExeQkJMXFxCj49csrKPx8ey0dHH1NTR1dXU
1gDV1tbW2dbW09nZp9PTR6enQkdHT0JCU09PS1NTRktLR0ZGT0dHWk9PY1paZmNj
YGZmVmBgO1ZWWTs7lFlZuJSUyri40srK1dLSANXV1tbW19fX3d3dwcHBbm5uOzs7
YmJiZ2dnX19fUFBQQkJC////////RkZGT09PWlpaYmJiZmZmYGBgNzc3cHBwo6Oj
wcHBz8/P1NTU1tYA2trW3d3ap6fdPDynWVk8hoZZgYGGbm6BVVVuQEBV//9A////
MTH/NzcxQkI3T09CYWFPZGRhTU1kT09NiYlPtraJysq209PK1gDT3dbWxt3dfMbG
Q3x8hUNDo4WFk6OjdpOTW3Z2SFtb/0hI////Kv//KyoqLysrOS8vUzk5aFNTZmho
NWZmczU1q3NzxKur0sTEANLS2NjY4uLira2tV1dXZGRktbW1t7e3nZ2df39/ZWVl
UlJS////////LCwsLi4uMDAwLy8vQEBAXl5eZmZmREREX19flJSUvb29z8/P2toA
4+Pan5/jTEyffX1MxcV9xcXFra3FjY2tcXGNWFhx//9Y////NDT/PDw0Pj48MzM+
NzczVlY3ZGRWSUlkWFhJiopYuLiKzs642gDO4tranOLiS5ychktLz4aG/8/P////
//////////////////////////////////////YP//SWBgVklJiVZWt4mJ
zbe3AM3N29vb5eXln5+fTExMiYmJ1tbW////////////////////////////////
//////////////////X19fSUlJVlZWiYmJt7e3zc3N29sA5eXbrKzlWFisd3dY
1dV33d3Vzs7dubnOoKC5fX2g//99////R0f/RERHODhELi44ODguUlI4YmJSQUFi
YGBBlJRgu7uUzs672gDO5NrawuTkccLCV3Fxx1dX38fH1t/fw9bWrMPDjKys/4yM
////Sv//Q0pKN0NDMjc3PzIyVT8/XVVVOV1dcDk5pHBwwaSk0cHBANHR2NjY4+Pj
2dnZkJCQRUVFqKio3d3d3t7ezc3NuLi4np6e/////////U1NTSEhIOTk5Nzc3RkZG
WVlZSkpKQ0NDhISEs7OzyMjI09PT2NgA4ODY5eXgtrblWVm2bW1Zzc1t4eHN1tbh
xcXWr6/F//+v////aWn/WVlpSEhZQUFITExBWFhMNTVYZmY1nZ1mvr6dzc2+1NTN
1gDU3NbW5tzc3ubmoN7eNaCgkjU11ZKS2dXVztnZvM7O/7y8////f///bH9/VGxs
SFRUT0hIQU9PQUFBi0FBs4uLx7Oz0sfH1tLSANbW1tbW2NjY39/f5+fn1dXVgYGB
GxsbioqKysrKz8/Pw8PDuLi4pqamkZGRfHx8Y2NjSkpKhoaGrq6uwsLC
zs7O1NTU1tbW1tYA1tbW2dnW39/Z4+Pf1NTjoqLURkaiWFhGi4tYpKSLqamknZ2p
iYmdcnKJUVFyNDRRTU00j49NsbGPwMCxzMzA0tLM1dXS1tbV1gDW1tbW1tbW2dbW
39nZ5d/f2uXlrdragq2tZIKCTWRkTE1NSkpMRkpKSUZGWUlJeVlZnHl5upycxLq6
y8vTE0cvL1NHR1tTU1tbWANbW1tbW1tbW1tbW2dnZ39/f4+Pj39/fyMjIq6urr
kJCQhoaGhISEhISEjIyMnJycjYyMtLS0LS0tcXFxy8vLzs7O0dHRHR0dPT09PT1dXV1tbW1tbW19fX1tYA
1tbW1tbW1tbW1tbW2NjW3NzY39/c4eHf3t7h1NTey8vUyMjLx8fIx8fHy8vHL
0dHP0dHR09PR1NTT1dXU1tbV1tbW19fWwDX19fX1tfX1tfX1tbW1tbW19bW
2tfX3dra3N3d2dzc19nZ1tfX1NbW1NTU1dTU1NXV1NTU1dTU1dXV1tXV1tbW19bW
19fXANfX
FORMAT=
QUERY=
RULE=
END
Object=0
Tag=0
Layer=0
Text=Sample1
Upper=2000.000000

```
 Lower=0.000000
 XPos=-80.000000
 YPos=45.000000
 Angle=0
 V=1
 E=0
 S=0
End
Object=1
 Tag=0
 Layer=2
 Text=Sample2
 Upper=2000.000000
 Lower=0.000000
 XPos=-80.000000
 YPos=47.000000
 Angle=0
 V=1
 E=0
 S=0
End
```

26. Appendix “C” - Sample UserCAD file listing

```
STYLE=-1
  NM=Style -1
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|FFFF00|1|0000FF|0|0
  BRUSH=0|FFFF00|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|0|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=
END
STYLE=0
  NM=Style 0
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=3|FF0000|1|0000FF|0|0
  BRUSH=0|008000|0000FF|27|0
  FONT=Arial|12|FF0000|FFFFFF|3|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|0|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=
END
STYLE=1
  NM=Style 1
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=12|00FFFF|1|0000FF|0|5036
  BRUSH=0|008000|0000FF|1|0
  FONT=Arial|28|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|252|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=
END
STYLE=2
  NM=Style 2
  VS=255
  EN=255
```

```
  UPPER=5000.000000
  LOWER=0.000000
  PEN=4|008000|1|0000FF|0|8844
  BRUSH=5|FF0000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|120|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=
END
STYLE=6
  NM=Style 6
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|6168
  BRUSH=5|000000|1FFFFFF|13|0
  FONT=Arial|-13|008000|FFFFFF|19|5|0|1|2.000000|0.000000|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|48|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=
END
OBJ:5:0:0:CAD Line
P:1
-73000000,40500000
END
OBJ:5:6:0:Labeling with CAD Object
P:1
-71600000,42200000
END
OBJ:3:6:1.000000:
P:2
-71600000,42000000
-71100000,42000000
END
OBJ:5:0:0:Testing the CAD Text 03
P:1
-80000000,42000000
END
OBJ:1:0:0:
P:2
-71600000,42400000
-71300000,42200000
END
OBJ:1:1:0:
P:2
-80500000,46000000
-70000000,41500000
END
```

OBJ:2:2:0:
P:2
-80000000,44000000
-79000000,45000000
END

27. Appendix “D” - Sample Theme .TRT file listing

```
THEME:World-Countries
  FILE:D:\DEVELOP\UNDERTOW\MapTivate\SAMPLE-SOURCE-
DELPHI\COUNTRY_COL_REGION.SHP
  NAME:World-Countries
  EXTENTS:-180.000000,-59.440900,180.000000,83.674733
  CONNECT:0
  VISIBLE:1
  ENABLED:1
  PRIORITY:0.0
  UPPER:5000.0
  LOWER:0.0
  FLAGS=10
  FILTER=
  MRURULES=0
STYLE=0
  NM=Style 0
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|6612
  BRUSH=0|008000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=Arial|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=
END
STYLE=1
  NM=AFGHANISTAN
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|5012
  BRUSH=0|FFFF00|0000FF|541937476|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=Arial|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AFGHANISTAN"
END
STYLE=2
  NM=ALBANIA
  VS=255
  EN=255
```

```
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|4104
  BRUSH=0|00FFFF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ALBANIA"
END
STYLE=3
  NM=ALGERIA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|12288
  BRUSH=0|FF00FF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ALGERIA"
END
STYLE=4
  NM=AMERICAN SAMOA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|11460
  BRUSH=0|FFFFFF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=Arial|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AMERICAN SAMOA"
END
STYLE=5
  NM=ANDORRA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|10572
  BRUSH=0|000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ANDORRA"
```

```
END
STYLE=6
  NM=ANGOLA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|32
  BRUSH=0|808080|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ANGOLA"
END
STYLE=7
  NM=ANGUILLA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|8860
  BRUSH=0|0000FF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ANGUILLA"
END
STYLE=8
  NM=ANTIGUA AND BARBUDA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|8024
  BRUSH=0|008000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ANTIGUA AND BARBUDA"
END
STYLE=9
  NM=ARGENTINA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|7048
  BRUSH=0|FF0000|0000FF|0|0
```

```
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ARGENTINA"
END
STYLE=10
  NM=ARMENIA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|6088
  BRUSH=0|FFFF00|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ARMENIA"
END
STYLE=11
  NM=ARUBA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|5208
  BRUSH=0|00FFFF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ARUBA"
END
STYLE=12
  NM=AUSTRALIA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|4332
  BRUSH=0|FF00FF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AUSTRALIA"
END
STYLE=13
  NM=AUSTRIA
  VS=255
```

```
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|13832
  BRUSH=0|FFFFFF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AUSTRIA"
 END
 STYLE=14
  NM=AZERBAIJAN
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|12880
  BRUSH=0|000000|0000FF|71589750|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=Arial|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AZERBAIJAN"
 END
 END
```

28. Appendix “E” - Sample Styles .TRS file listing

```
STYLE=0
  NM=Style 0
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|6612
  BRUSH=0|008000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=Arial|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=
END
STYLE=1
  NM=AFGHANISTAN
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|5012
  BRUSH=0|FFFF00|0000FF|541937476|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=Arial|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AFGHANISTAN"
END
STYLE=2
  NM=ALBANIA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|4104
  BRUSH=0|00FFFF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ALBANIA"
END
STYLE=3
  NM=ALGERIA
  VS=255
  EN=255
```

```
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|12288
  BRUSH=0|FF00FF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ALGERIA"
END
STYLE=4
  NM=AMERICAN SAMOA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|11460
  BRUSH=0|FFFFFF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AMERICAN SAMOA"
END
STYLE=5
  NM=ANDORRA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|10572
  BRUSH=0|000000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ANDORRA"
END
STYLE=6
  NM=ANGOLA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|32
  BRUSH=0|808080|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ANGOLA"
```

```
END
STYLE=7
  NM=ANGUILLA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|8860
  BRUSH=0|0000FF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ANGUILLA"
END
STYLE=8
  NM=ANTIGUA AND BARBUDA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|8024
  BRUSH=0|008000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ANTIGUA AND BARBUDA"
END
STYLE=9
  NM=ARGENTINA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|7048
  BRUSH=0|FF0000|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ARGENTINA"
END
STYLE=10
  NM=ARMENIA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|6088
  BRUSH=0|FFFF00|0000FF|0|0
```

```
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ARMENIA"
END
STYLE=11
  NM=ARUBA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|5208
  BRUSH=0|00FFFF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="ARUBA"
END
STYLE=12
  NM=AUSTRALIA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|4332
  BRUSH=0|FF00FF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AUSTRALIA"
END
STYLE=13
  NM=AUSTRIA
  VS=255
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|13832
  BRUSH=0|FFFFFF|0000FF|0|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=ARIAL|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AUSTRIA"
END
STYLE=14
  NM=AZERBAIJAN
  VS=255
```

```
  EN=255
  UPPER=5000.000000
  LOWER=0.000000
  PEN=1|0000FF|1|0000FF|0|12880
  BRUSH=0|000000|0000FF|71589750|0
  FONT=Arial|8|000000|FFFFFF|0|5|0|1|2.000000|0.000000|0|0|0|0|0|0|0
  MARK=Arial|64|8|1|0000FF|FF0000|1|1|2000.000000|0.000000
  FORMAT=
  QUERY=
  RULE=F0="AZERBAIJAN"
END
```