XML Traffic Radar 2.0.1

by Arne Bartels arne.bartels@t-online.de

NOTE

The previously uploaded version had a serious bug, so on some (most) systems TrafficRadarXML.dll wasn't loaded. Please overwrite with this version.

NOTE

Introduction:

This package contains a module dll that can be used as traffic radar or TCAS in XML gauges for FS9. Source code is also provided.

Use the included XML gauges directly in a panel, modify it, or use the source to build a TCAS-like gauge.

The source code is meant as an example to C++ coding, XML to C interface, the FS-traffic interface (ITrafficInfo) and GDI+ programming.

Direct use of the XML gauge needs minimal knowledge of gauge placement, the XML code needs general knowledge of XML gauge coding, the C++ code requires a good knowledge of C++ programming.

The TrafficRadar XML-gauge utilizes a radar-like screen showing the surrounding aircraft in different colors, aircraft on ground are in white, flying in green, aircrafts very near in two different colors; in TA range in yellow, in RA range in red. Range is selected on the lower two corners, brightness is selected in the top two corners, placing the cursor over an aircraft symbol displays aircraft info in the tooltip. The panel circuit has to be on, to make the display work.

The TAS600MHD XML-gauge looks and works a bit like the Multi-Hazard Display for Avidyne's TAS600 Traffic Advisory System (see http://www.avidyne.com/products/tas600_overview.shtm or http://www.avidyne.com/products/tas600_brochure.shtm). It features the typical look of TCAS displays, with different symbols for different categories. Blue hollow diamonds are aircraft „other" traffic, not near your own plane, blue filled diamonds show „proximity" traffic, yellow filled circles „traffic advisories". Brightness and range are controlled via the BRT and RNG knobs, all other buttons are non-functional.

A replacement for the standard gps_500.xml gauge adds a TCAS layer to the existing GPS gauge.

Installation:
put TrafficRadarXML.dll from Modules in the modules folder of FS9, the „Radar" folder from „Gauges" in the gauges folder of FS9. If you want to use the replacement GPS place the „fs9gps"-folder in the gauges sub folder. If you have uncapped the fs9gps.cab before, backup the original gps_500.xml file, and replace it with the one from this package (gauges\fs9gps\gps_500.xml). But if you have, you know already what to do.

„Source" contains the source code for the dll, if you don't need it, ignore it.

Incorporate the XML gauges in a panel.cfg the usual way, e.g.

gauge52=Radar!TrafficRadar,200,0,200,200
gauge53=Radar!TAS600MHD,400,0,200,200

You may need the TrafficInfo.dll from the Panel SDK of MS (http://www.microsoft.com/games/flightsimulator/fs2004_downloads_sdk.asp#panels) or you move it from the packages Modules folder to the FS Modules folder.
On older sytems (e.g. Win98,Win2000) the gdiplus.dll is needed, get it from MS and place it in the FS9 folder (e.g. http://www.microsoft.com/downloads/details.aspx?FamilyID=6a63ab9c-df12-4d41-933c-be590feaa05a&DisplayLang=en).

XML code:
Two different <CustomDraw> elements are available: fs9traffic:radar shows a radar-style screen, fs9traffic:tcas looks more like a real TCAS display. They differ in the default settings, see also the table further down. The possible parameters are exactly the same, though.

Insert in a XML gauge like e.g.:

```
<Element>
   <CustomDraw Name="fs9traffic:radar" X="100" Y="100"  >
    <Heading>(A:GPS GROUND TRUE HEADING, degrees)</Heading>
    <Range>(C:fs9traffic:range,number) </Range>
    <Brightness>(C:fs9traffic:brightness,number)</Brightness>
   </CustomDraw>
</Element>
```

There are a vast amount of modification possibilities available, play around a bit.
Please note that all colours have to be in hexadecimal format. Also no <CustomDraw> element evaluates an expression in an attribute, use element syntax instead, e.g.

```
<Element>
  <CustomDraw Name="fs9traffic:radar" X="100" Y="100">
  <ProximityColour>'0x5fffaa' '0x7ac688' '0x7ab678' 3 (L:SColor,enum) case
</ProximityColour>
```

```
  <Heading>(A:GPS GROUND TRUE HEADING, degrees)</Heading>
    <Range>(C:fs9traffic:range,number) </Range>
    <Brightness>(C:fs9traffic:brightness,number)</Brightness>
 </CustomDraw>
</Element>
```

The Syntax for the aicraft info is abit more complex, the display is easy:

```
<Tooltip>%((C:fs9traffic:toolTip,string))%!s!%</Tooltip>
```

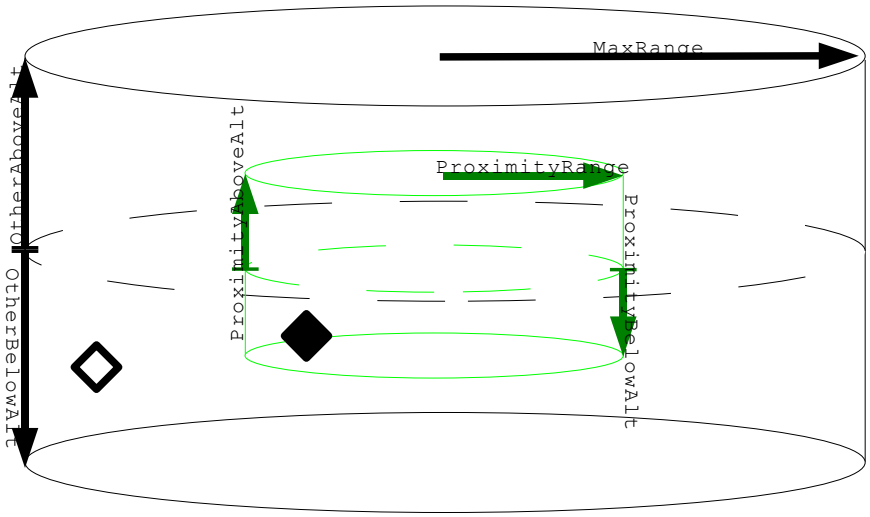
To work properly it is also needed to broadcast the mouse position back to the dll. The variables `C:fs9traffic:mouseRelPosX` and `C:fs9traffic:mouseRelPosY` take a relative mouse position. If the mouse is in the top left corner of the view screen the position is 0,0 in the bottom right corner it is 1,1. The mouse position has to be scaled to fit. In the „TrafficRadar“ gauge code, it is done by using an <Area> of the same size and position as the <CustomDraw>. Mouse X- and Y-Position are scaled by the width and height before written to the Variables.
You can even „roll your own“ Tooltip by using the `(C:fs9traffic:toolTip...)` variables. The default Tooltip could also be written with:

```
<Tooltip>%((C:fs9traffic:toolTipId,string) d )%{if}%ID: %!s! ALT:
%((C:fs9traffic:toolTipAltDiff,number) 100 / )%!d!00ft GS:
%((C:fs9traffic:toolTipGroundSpeed,number))%!d!kt TRK:
%((C:fs9traffic:toolTipTrackAngle,number))%!d!&#176; DST:
%((C:fs9traffic:toolTipDist,number))%!.1f!nm%{end}</Tooltip>
```

Please note that all C: variables are local to the XML gauge they are defined in, much like G: variables. If you have controls on other gauges, you can't use the `C:fs9traffic` variables, because the <CustomDraw>s on other gauges won't notice changes; use L: vars instead. Especially `C:fs9traffic:range and C:fs9traffic:brightness` are only helper variables so you don't have start values of 0, no one stops you from using your own variables for each of the possible parameters.

If you change the „Font“, do it with care, it might result in crashes if the font isn't available.

What are these strange „categories“? On TCAS-systems it is usual to have „other“ traffic aircrafts within a certain height above or below your plane and within the total range of the TCAS, often displayed as hollow white diamonds (OtherAboveAlt,OtherBelowAlt,MaxRange). More near to your aircraft is the „proximity“ traffic usually displayed as white filled diamonds (ProximityAboveAlt, ProximityBelowAlt, ProximityRange). „Other“ traffic can include aircraft on/near ground (OtherMinRAlt=0), „proximity“ traffic includes only aircraft above ground.



Aircrafts too near to your aircraft are shown as Traffic Advisory (TA), usually displayed as yellow circles. It means that evasive action might be needed shortly. Resolution Advisories (RA) means immediate action is needed to avoid collisions, shown as red squares. However a „real“ RA provides includes vertical commands, this TCAS can't, it only gives the warning not the needed

action.

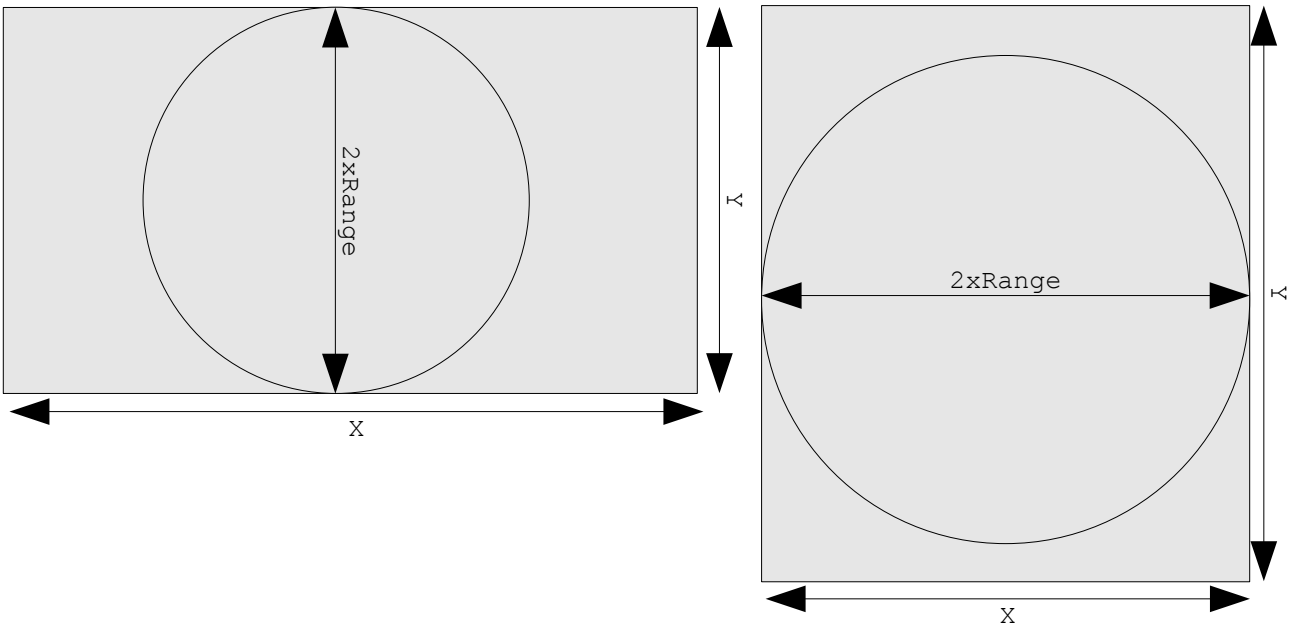
The `fs9traffic:tcas` uses already pretty good presets for average TCAS's. `fs9traffic:radar` takes the parameters mentioned above to the extreme, so that all airaft are displayed (huge values Above/Below, same ProximityRange as MaxRange), all aircraft should be “proximity” then, but with OtherMinRAlt=0 “other” includes aircraft on ground, so “other” are the aircraft on ground, “proximity” all in air.

To determine a TA or RA an algorithm in three parts is used:

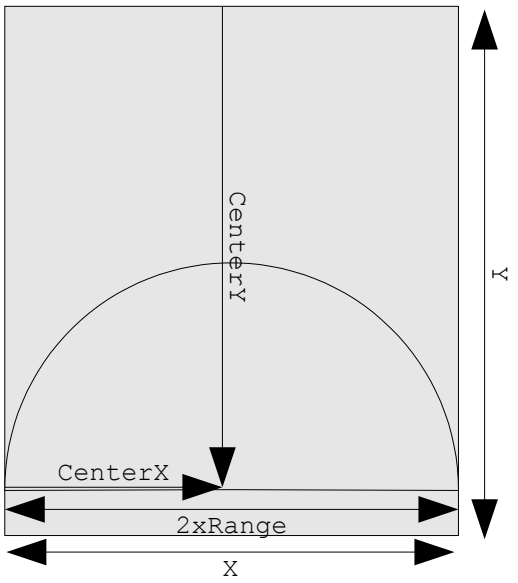
Each second all aircraft within “MaxRange” up to “MaxNumAct” are read from FS. From the slant distance at the previous cycle, the change of distance gives the proximity speed. With this and the present distance a time can be calculated. This time is the time to the closest point of approach, or in other words the time you have to prevent a collision. If this time is below `C:fs9traffic:timeSeparationTA` a TA might be issued. Additionally, if the proximity speed is low, there should at least a horizontal distance of at least `C:fs9traffic:distSeparationTA`.

To avoid unnecessary alerts TAs are only issued if also the altitude difference is smaller then `C:fs9traffic:altSeparationTA` during the time till the closest point of approach. Similar logic is valid for RAs. These 2x3 variables are deliberately chosen to be write able: different aircraft use different sets of parameters, from time to time the values are changed to fit newer regulations. E.g. the TAS600MHD example can't issue RAs, therefore the `C:fs9traffic:..RA` values are set to 0, suppressing all RAs. The TrafficRadar example shows a more sophisticated way of calculating the separation values, fit for big airliners. See also http://www.aerowinx.com/html/tcas.html for description of terms, algorithms and values.

For square gauges with a centered pivot point, the use of „`Range`“ is clear, but what if the drawing area is a rectangle? Then the shorter side represents two times „`Range`“.



This is even true for eccentric gauges:



The fs9gps gauge uses a fixed resolution independent of gauge size and gauge aspect ratio: the symbols have always the same size. Set `FixedResolution="1"` then.

Due to the Anti Aliasing, the symbols might look bad on light colored backgrounds, like the terrain map of the GPS, you can switch it off (`AntiAliasing="0"`) and/or pad each symbol with a dark grey background rectangle (`PadSymbols="1"`).

The `..Detail` Parameters determine how the symbols look like, you can combine 1 (dot), 2 (outlined symbol), 4 (filled symbol), 8 (difference altitude) and 16 (ATC ID) to a bit sum.

With the great amount of possible parameters shown below, a lot of variants are thinkable.

- For a ground (taxi) radar suppress all aircraft in air by setting ProximityDetail, TADetail, RADetail to 0.
- The other way round, all aircraft in air none on ground: OtherDetail="0" or OtherMinRAlt to a value>0.
- All flying aircraft green (a military radar?) set all ..Color values to green.
- Same as above, but all advisories as filled circles: all ..Color values to green, TADetail and RADetail to 14 (outlined+filled+altitude)
- Real world TCASs allow to choose display of TA only TA/RA only, do this by switching the not displayed ..Detail values to 0.
- A tooltip with radar altitude instead of difference altitude?
  Use:
```
<Tooltip>%((C:fs9traffic:toolTipId,string) d )%{if}%ID: %!s! RADALT:
%((C:fs9traffic:toolTipRadAlt,number) )%!d!ft GS:
%((C:fs9traffic:toolTipGroundSpeed,number))%!d!kt TRK:
%((C:fs9traffic:toolTipTrackAngle,number))%!d!&#176; DST:
%((C:fs9traffic:toolTipDist,number))%!.1f!nm%{end}</Tooltip>
```
- You can even use two `fs9traffic:` <CustomDraw>s in one gauge together eg something like the Garmin GNS530 with a small secondary screen at the bottom left only for TCAS purposes with less detail (OtherDetail to 2 all other ...Detail to 4 => symbol only) see also

http://www.garmin.com/products/gns530/hi.html.

If you update from the previous version please note some differences:

replace „C:TrafficRadar:“ „C:fs9traffic:“ in your XML-gauges. Note that the former „TrafficRadar:TCAS“ has to be replaced by „fs9traffic:radar”. All “Inactive...” parameters are replaced by “Other..” and “Standard...” by “Proximity..”. Change from Radar!TrafficRadarXML to Radar!TrafficRadar in panel.cfgs. For backward compatibility the „old“ names still work though.

Available elements:

CustomDraw	Property ID
fs9traffic:radar	0
fs9traffic:tcas	1

C:variables	Comment/Description	Default	Property ID
C:fs9traffic:range	Preset range	15nm	2
C:fs9traffic:brightness	Preset brightness	255	3
C:fs9traffic:mouseRelPosX	Relative position of mouse pointer 0.0 is left 1.0 right	0	4
C:fs9traffic:mouseRelPosY	Relative position of mouse pointer 0.0 is top 1.0 bottom	0	5
C:fs9traffic:toolTip	ATC ID,altitude difference, ground speed, ground track, distance at mouse position	(read-only)	6
C:fs9traffic:advisory	0: no advisory, 1: traffic advisory active,2: resolution advisory active	(read-only)	7
C:fs9traffic:timeSeparationTA		35s	8
C:fs9traffic:timeSeparationRA		20s	9
C:fs9traffic:distSeparationTA		1nm	10
C:fs9traffic:distSeparationRA		0.1nm	11
C:fs9traffic:altSeparationTA		1200ft	12
C:fs9traffic:altSeparationRA		300ft	13
C:fs9traffic:toolTipId	ATC ID at mouse position	(read-only)	14
C:fs9traffic:toolTipAlt	altitude at mouse position (ft)	(read-only)	15
C:fs9traffic:toolTipRadAlt	Radar altitude at mouse position (ft)	(read-only) not for MP	28
C:fs9traffic:toolTipDist	distance at mouse position (nm)	(read-only)	16
C:fs9traffic:toolTipBrg	Bearing due true north at mouse position (°)	(read-only)	17
C:fs9traffic:toolTipAltDiff	altitude diff at mouse position (ft)	(read-only)	18
C:fs9traffic:toolTipTrackAngle	ground track at mouse position (°)	(read-only)	19
C:fs9traffic:toolTipGroundSpeed	ground speed at mouse position (kts)	(read-only)	20
C:fs9traffic:advisoryId	ATC ID advisory	(read-only)	21
C:fs9traffic:advisoryAlt	altitude advisory (ft)	(read-only)	22
C:fs9traffic:advisoryRadAlt	Radar altitude advisory (ft)	(read-only)	23

		not for MP	
C:fs9traffic:advisoryDist	distance advisory (nm)	(read-only)	24
C:fs9traffic:advisoryBrg	Bearing due true north advisory (°)	(read-only)	25
C:fs9traffic:advisoryAltDiff	altitude diff advisory (ft)	(read-only)	26
C:fs9traffic:advisoryTrackAngle	ground track advisory (°)	(read-only)	27
C:fs9traffic:advisoryGroundSpeed	ground speed advisory (kts)	(read-only)	29

Possible parameters for both <CustomDraw>s.

DrawProperty	Comment/Description	Default radar	Default tcas	ID
Range	Equivalent to the smaller of width and height	C:fs9traffic:range	C:fs9traffic:range	0
Heading	0..360°	0	0	1
Brightness	0..255	C:fs9traffic:brightness	C:fs9traffic:brightness	2
OtherColour	Colors always hexadecimal: 0xRRGGBB	0xF0F0F0	0xF0F0F0	3
ProximityColour		0x00FF00	0xF0F0F0	4
TAColour		0xFFFF00	0xFFFF00	5
RAColour		0xFF0000	0xFF0000	6
AntiAlias	May look bad on light backgrounds, see also PadSymbols	1	1	7
MaxRange	Maximal distance of polled aircrafts	100	40	8
MaxNumAct	Maximal number of polled aircraft	100	100	9
CenterX	Center of Rotation	X/2	X/2	10
CenterY		Y/2	Y/2	11
SymbolSize	Scale factor for symbols	1	1	12
OtherDetail	Bit sum of: – 0 not displayed – 1 dot – 2 outlined symbol – 4 filled symbol – 8 altitude string – 16 id string	10	10	13
ProximityDetail		10	14	14
TADetail		10	14	15
RADetail		10	14	16
FixedResolution	resolution independent of size, needed for fs9_gps	0	0	17
PadSymbols	Pad symbols with dark grey background, for light backgrounds	0	0	18
Font	Change with care, can cause CTDs!	Arial	Arial	19
OtherMinRAlt	Minimum radar altitude for „other aircrafts“	0 (aircraft on ground	50ft (aircraft on ground not	20

		included)	included)	
OtherAboveAlt	Maximum altitude difference for „other aircraft“ above own aircraft	100000ft (all aicraft above)	2700ft	21
OtherBelowAlt	Maximum altitude difference for „other aircraft“ below own aircraft	100000ft (all aicraft below)	2700ft	22
ProximityRange	Maximum range for „proximity aircraft“	MaxRange (all polled aircraft)	6nm	23
ProximityAboveAlt	Maximum altitude difference for „proximity aircraft“ below own aircraft	100000ft (all aicraft above)	1200ft	24
ProximityBelowAlt	Maximum altitude difference for „proximity aircraft“ above own aircraft	100000ft (all aicraft below)	1200ft	25
MaxDispNumAct	Maximal number of displayed aircraft symbols	MaxNumAct	MaxNumAct	26

C++ code:
Use Microsoft Visual C++ 2005 for compiling the source code. For backward compatibility to older VC version some #defines are added in TrafficData.h. If your compiler supports the more secure ..._s variants, but has a lower compiler version, adjust `#if (_MSC_VER<1400)` to your needs.
For compilation create a dll project, add all .cpp, .rc files from sources.zip to the project, link user32.lib and gdiplus.lib (similar to .gau projects).

CAUTION

Make sure to use proper settings in Properties->C/C++->Code Generation->Runtime Library. /MT for release and /MTd for debug versions.

CAUTION

To understand the code please read the comments. Some of the code simply follows the necessary approach to build XML<->C dlls (see also XMLCinterface.zip by he same author), some code follows the necessary approach to use the ITrafficInfo-Interface and GDI+.
To allow strict local variables all calls to c:fs9traffic:... variables are calls to MyGaugeCCallback member variables. To allow two different CustomDraws with the exact same set of variables but different presets, a abstract class of MyGaugeCDrawable is used where all necessary code is placed in. The two child classes RadarCDrawable and TCASCDrawable just allocate different drawing classes and preset the same variables differently. Also for drawing the traffic an abstract class GdiPlusRadar does all necessary performance, only the actual drawing of the symbols is implemented in the child classes RadarScreen, TCASScreen. Note that the drawing of the symbol backgrounds is used to determine the maximum number of displayed aircraft, even if the symbol background isn't even drawn. From TrafficData the AI data is sorted from small distances to big distances, for determination of max number of drawn symbols (and for background padding) the array is run through forward, whereas for drawing the symbols, it has to be accessed backwards, so that the nearest aircraft (potentially more dangerous) overwrite the more distanced ones.
For backward compatibility the MyPanelCCallback class is allocated twice in TrafficRadarXML.cpp. In compability mode „`C:TrafficRadar:`“ is also recognized, where „`C:TrafficRadar:TCAS`“ is mapped to „`C:fs9traffic:radar`“.

Please note that <CustomDraw> elements can also be used by C-gauges, but this isn't tested with this dll.

This document and the associated source code is put in the public domain
Arne Bartels, Husum Germany, July 2006