

Rave Reports 5.1 Scripting Documentation

Rave Event Scripting

The Event Editor in Rave allows for the definition of several types of events components within the Rave visual report environment. This can be used to extend the existing functionality of the visual components with custom logic for much more flexible reports. This document describes the basics of the Rave scripting language and the event editor. For examples of several reports that use scripting, view the reporting project located in the Rave5\Scripting folder.

The Rave Event Editor



- 1) Event Editor Tab – Select this tab to view the event editor within the Rave report designer.
- 2) Available Events dropdown list – This control shows the current list of available events for the currently selected component. When an event is selected in this list, it is then created and moved to the Defined Events list. This control is disabled when the Event Editor is in page view mode.
- 3) Defined Events list – This list shows all of the defined events for the currently selected component (if in selected mode) or for all of the components on the currently selected page (if in page view mode). Selecting different events on this list will change which event is visible in the lower portion of the Event Editor.
- 4) Event Description – This section will display helpful information on the parameters and usage of the currently selected event.
- 5) Compile button – This will perform a compile of all events in the currently loaded reporting project. It is not necessary to hit this button each time a change is made as Rave will automatically compile if necessary before a report is run within the designer.
- 6) Show Events for Selected button – When this button is selected the events displayed in the Defined Events list will be those for the currently selected component (use the Project Tree to change selection).
- 7) Show All Events on Page button – When this button is selected the events displayed in the Defined Events list will be those for all components on the currently selected page (use the Project Tree to change selection). When in this mode, the Available Events dropdown list will be disabled.
- 8) Delete Current Event button – This button can be used to delete an event that is no longer needed.

- 9) Compile Status – This label will display the results of the most recent compile.
- 10) Event declaration text – This read-only text will show the actual declaration for the current event.
- 11) Event header memo – This memo will allow for the definition of variables and other non-code items for the current event.
- 12) Event body memo – This memo is where code statements for the current event should be placed.

Scripting Syntax guide

If statement

```
if condition then  
    statement(s);  
elseif condition then  
    statement(s);  
else  
    statement(s);  
end;
```

For loop

```
for index := start to finish do  
    statement(s);  
end;  
  
for index := start downto finish do  
    statement(s);  
end;
```

While loop

```
while condition do  
    statement(s);  
end;
```

Repeat loop

```
repeat  
    statement(s);  
until condition;
```

With statement

```
with object do  
    statement(s);  
end;
```

Break, Continue and Exit function similar to Delphi.

Variables may be declared similar to Delphi. The following types are supported:

Boolean, Shortint (-128 to 127), Byte (0 to 255), SmallInt (-32768 to 32767), Word (0 to 65535), Integer (32 bit signed integer), Single (4 byte float), Double (8 byte float), Extended, Currency, Char and String.

Standard functions

```
function ShowMessage(S1: string);
function IntToStr(I1: integer): string;
function StrToInt(S1: string): Integer;
function StrToIntDef(S1: string; Default: Integer): Integer;
function StrToFloat(S1: string): Extended;
function FloatToStr(E1: extended): string;
function Copy(S1: string; Index, Count: integer): string;
function Delete(var S1: string; Index, Count: Integer);
function Insert(Source: string; var S1: string; Index: integer);
function Length(S1: string): Integer;
function Pos(Substr: string; S1: string): Integer;
function Compare(S1, S2: string): Integer;
function CompareCase(S1, S2: string): Integer;
function Trim(S1: string): string;
function TrimLeft(S1: string): string;
function TrimRight(S1: string): string;
function MakeStr(Ch: char; Count: integer): string;
function UpperCase(S1: string): string;
function LowerCase(S1: string): string;
function Chr(X: byte): char;
function CurrToStr(Value: currency): string;
function StrToCurr(S1: string): currency;
function DateTimeToStr(DateTime: TDateTime): string;
function StrToDateTime(S1: string): TDateTime;
function DateToStr(Date: TDateTime): string;
function StrToDate(S1: string): TDateTime;
function TimeToStr(Time: TDateTime): string;
function StrToTime(S1: string): TDateTime;
function Abs(X: integer): integer;
function AbsExt(X: extended): extended;
function ArcTan(X: extended): extended;
function Cos(X: extended): extended;
function Sin(X: extended): extended;
function Exp(X: extended): extended;
function Frac(X: extended): extended;
function Ln(X: extended): extended;
function Round(X: extended): integer;
function Trunc(X: extended): integer;
function Odd(X: integer): boolean;
function Pi: extended;
function Random(Range: integer): integer;
function Randomize;
function Sqr(X: extended): extended;
function Sqrt(X: extended): extended;
function Beep;
function Date: TDateTime;
function DayOfWeek(Date: TDateTime): integer;
function BreakupDate(Date: TDateTime; var Year, Month, Day: word);
function BreakupTime(Time: TDateTime; var Hour, Min, Sec, MSec: word);
function CreateDate(Year, Month, Day: word): TDateTime;
function CreateTime(Hour, Min, Sec, MSec: word): TDateTime;
function Now: TDateTime;
function QuotedStr(Value: string; Quote: char): string;
```

Standard Classes

Please refer to the *.RVS files in the Rave5\Source folder for which properties and methods of the Rave classes are available. These source files are written in Rave script and are compiled with the RaveCC compiler to produce the .RVC files that are used directly by Rave.