Mode: Differences with Context

Left file: D:\My documents\Delphi 3rd party components\jvcl\run\JvClock_v128.pas
Right file: D:\My documents\Delphi 3rd party components\jvcl\run\JvClock.pas

56	FTimer: TJvTimer;	=	56	FTimer: TJvTimer;
57	FAutoSize: Boolean;		57	FAutoSize: Boolean;
58	FShowMode: TShowClock;		58	FShowMode: TShowClock;
59	FTwelveHour: Boolean;		59	FTwelveHour: Boolean;
60	FLeadingZero: Boolean;		60	FLeadingZero: Boolean;
61	FShowSeconds: Boolean;		61	FShowSeconds: Boolean;
		-+	62	FClockEnabled: Boolean;
			63	FFixedTime: TDateTime;
62	FAlarm: TDateTime;	=	64	FAlarm: TDateTime;
63	FAlarmEnabled: Boolean;		65	FAlarmEnabled: Boolean;
64	{$IFDEF VCL}		66	{$IFDEF VCL}
65	FHooked: Boolean;		67	FHooked: Boolean;
66	{$ENDIF VCL}		68	{$ENDIF VCL}
67	FDotsColor: TColor;		69	FDotsColor: TColor;

76	FOnGetDate: TJvGetDateEvent;	=	78	FOnGetDate: TJvGetDateEvent;
77	FDateFormat: string;		79	FDateFormat: string;
78	procedure TimerExpired(Sender: TObje » ct);		80	procedure TimerExpired(Sender: TObje » ct);
79	procedure GetTime(var T: TJvClockTim » e);		81	procedure GetTime(var T: TJvClockTim » e);
80	function IsAlarmTime(ATime: TDateTim » e): Boolean;		82	function IsAlarmTime(ATime: TDateTim » e): Boolean;
81	procedure SetShowMode(Value: TShowCl » ock);		83	procedure SetShowMode(Value: TShowCl » ock);
		-+	84	procedure SetClockEnabled(Value: Boo » lean);
			85	procedure SetFixedTime(Value: TDateT » ime);
82	function GetAlarmElement(Index: Inte » ger): Byte;	=	86	function GetAlarmElement(Index: Inte » ger): Byte;
83	procedure SetAlarmElement(Index: Int » eger; Value: Byte);		87	procedure SetAlarmElement(Index: Int » eger; Value: Byte);
84	procedure SetDotsColor(Value: TColor »);		88	procedure SetDotsColor(Value: TColor »);
85	procedure SetTwelveHour(Value: Boole » an);		89	procedure SetTwelveHour(Value: Boole » an);
86	procedure SetLeadingZero(Value: Bool » ean);		90	procedure SetLeadingZero(Value: Bool » ean);
87	procedure SetShowSeconds(Value: Bool » ean);		91	procedure SetShowSeconds(Value: Bool » ean);

119	public	=	123	public
120	constructor Create(AOwner: TComponen » t); override;		124	constructor Create(AOwner: TComponen » t); override;
121	destructor Destroy; override;		125	destructor Destroy; override;
122	procedure SetAlarmTime(AlarmTime: TD » ateTime);		126	procedure SetAlarmTime(AlarmTime: TD » ateTime);
123	procedure UpdateClock;		127	procedure UpdateClock;
124	published		128	published
		-+	129	property ClockEnabled: Boolean read » FClockEnabled write SetClockEnabled de » fault True;
			130	property FixedTime: TDateTime read F » FixedTime write SetFixedTime;
125	property AlarmEnabled: Boolean read » FAlarmEnabled write FAlarmEnabled defa	=	131	property AlarmEnabled: Boolean read » FAlarmEnabled write FAlarmEnabled defa

Left file: D:\My documents\Delphi 3rd party components\jvcl\run\JvClock_v128.pas
Right file: D:\My documents\Delphi 3rd party components\jvcl\run\JvClock.pas
(continued)

	» ult False;			» ult False;
126	property AlarmHour: Byte index 1 rea		132	property AlarmHour: Byte index 1 rea
	» d GetAlarmElement write SetAlarmElemen			» d GetAlarmElement write SetAlarmElemen
	» t default 0;			» t default 0;
127	property AlarmMinute: Byte index 2 r		133	property AlarmMinute: Byte index 2 r
	» ead GetAlarmElement write SetAlarmElem			» ead GetAlarmElement write SetAlarmElem
	» ent default 0;			» ent default 0;
128	property AlarmSecond: Byte index 3 r		134	property AlarmSecond: Byte index 3 r
	» ead GetAlarmElement write SetAlarmElem			» ead GetAlarmElement write SetAlarmElem
	» ent default 0;			» ent default 0;
129	property AutoSize: Boolean read FAut		135	property AutoSize: Boolean read FAut
	» oSize write SetAutoSize default False;			» oSize write SetAutoSize default False;
130	property BevelInner default bvLowere		136	property BevelInner default bvLowere
	» d;			» d;
395	ControlStyle := ControlStyle - [csSetC	=	401	ControlStyle := ControlStyle - [csSetC
	» aption] - [csReplicatable];			» aption] - [csReplicatable];
396	{$IFDEF VCL}		402	{$IFDEF VCL}
397	IncludeThemeStyle(Self, [csNeedsBorder		403	IncludeThemeStyle(Self, [csNeedsBorder
	» Paint, csParentBackground]);			» Paint, csParentBackground]);
398	{$ENDIF VCL}		404	{$ENDIF VCL}
399	BevelInner := bvLowered;		405	BevelInner := bvLowered;
400	BevelOuter := bvRaised;		406	BevelOuter := bvRaised;
		-+	407	FClockEnabled := True;
401	FTimer := TJvTimer.Create(Self);	=	408	FTimer := TJvTimer.Create(Self);
402	FTimer.Interval := 450; { every second		409	FTimer.Interval := 450; { every second
»	}			» }
403	FTimer.OnTimer := TimerExpired;		410	FTimer.OnTimer := TimerExpired;
		-+	411	FTimer.Enabled := FClockEnabled;
404	FDotsColor := clTeal;	=	412	FDotsColor := clTeal;
405	FShowSeconds := True;		413	FShowSeconds := True;
406	FLeadingZero := True;		414	FLeadingZero := True;
407	FShowDate := False;		415	FShowDate := False;
408	FDateFormat := ShortDateFormat;		416	FDateFormat := ShortDateFormat;
409	GetTime(FDisplayTime);		417	GetTime(FDisplayTime);
498	end;	=	506	end;
499			507	
500	{$ENDIF VCL}		508	{$ENDIF VCL}
501			509	
502	function TJvClock.GetSystemTime: TDateTi		510	function TJvClock.GetSystemTime: TDateTi
	» me;			» me;
503	begin		511	begin
		<>	512	If FTimer.Enabled then
504	Result := SysUtils.Time;		513	Result := SysUtils.Time
			514	else
			515	Result := FFixedTime;
505	if Assigned(FOnGetTime) then	=	516	if Assigned(FOnGetTime) then
506	FOnGetTime(Self, Result);		517	FOnGetTime(Self, Result);
507	end;		518	end;
508			519	
509	function TJvClock.GetSystemDate: TDateTi		520	function TJvClock.GetSystemDate: TDateTi
	» me;			» me;
510	begin		521	begin
515		=	526	
516	procedure TJvClock.GetTime(var T: TJvClo		527	procedure TJvClock.GetTime(var T: TJvClo
	» ckTime);			» ckTime);
517	var		528	var

Left file: D:\My documents\Delphi 3rd party components\jvcl\run\JvClock_v128.pas
Right file: D:\My documents\Delphi 3rd party components\jvcl\run\JvClock.pas
(continued)

518	MSec: Word;		529	MSec: Word;
519	begin		530	begin
520	with T do		531	with T do

521	DecodeTime(GetSystemTime, Hour, Minu » te, Second, MSec);	<>	532
			533
			534
			535
			536
			537
				begin
				if FTimer.Enabled then
				DecodeTime(GetSystemTime, Hour, » Minute, Second, MSec)
				else
				DecodeTime(FFixedTime, Hour, Min » ute, Second, MSec)
				end;

522	end;	=	538	end;
523			539	
524	procedure TJvClock.UpdateClock;		540	procedure TJvClock.UpdateClock;
525	begin		541	begin
526	Invalidate;		542	Invalidate;
527	if AutoSize then		543	if AutoSize then

603	procedure TJvClock.Alarm;	=	619	procedure TJvClock.Alarm;
604	begin		620	begin
605	if Assigned(FOnAlarm) then		621	if Assigned(FOnAlarm) then
606	FOnAlarm(Self);		622	FOnAlarm(Self);
607	end;		623	end;
608			624	
		<>	625	procedure TJvClock.SetClockEnabled(Value » : Boolean);
			626	begin
			627	FClockEnabled := Value;
			628	FTimer.Enabled := FClockEnabled;
			629	if not FClockEnabled then
			630	SetFixedTime(Now);
			631	end;
			632	
			633	procedure TJvClock.SetFixedTime(Value: T » DateTime);
			634	begin
			635	FClockEnabled := False;
			636	FTimer.Enabled := False;
			637	FFixedTime := Value;
			638	TimerExpired(Self);
			639	end;
			640	
609	procedure TJvClock.SetAutoSize(Value: Bo » olean);	=	641	procedure TJvClock.SetAutoSize(Value: Bo » olean);
610	begin		642	begin
611	{$IFDEF VCL}		643	{$IFDEF VCL}
612	{$IFDEF COMPILER6_UP}		644	{$IFDEF COMPILER6_UP}
613	inherited SetAutoSize(Value);		645	inherited SetAutoSize(Value);
614	{$ENDIF COMPILER6_UP}		646	{$ENDIF COMPILER6_UP}