# TrueRAD Suite

## Units

radconlist
rade2m
rade2p
radp2p
radvar

## Classes

TradConnection
TradConnectionList
TradE2MConnection
TradE2PConnection
TradP2PConnection
TradParameterConnection
TradParameterConnections
TradVariable
TradVariableClassInfo

## Types

TradValueOptions

# String Handling Routines

Various utility routines/method/classes for string handling

**Description**
No description yet...

# File/Directory Name Manipulation Routines

Various utility routines/method/classes

## Components

## Buttons

## Labels

**Legend**

 - Marks that the item has an associated example. (this bitmap is a hyperlink.)

 - Marks that the item has documented bugs.

# radconlist Unit {button &Top,JI(`',`IDH_Unit_radconlist')}{button &Classes,JI(`',`IDH_UnitTopic_radconlist_Classes')} {button &Types,JI(`',`IDH_UnitTopic_radconlist_OtherTypes')}

Legend

**Classes**

**TradConnection**

TradConnection is the common ancestor of all connection components.

**TradConnectionList**

TradConnectionsList is the generic class for creating components that maintain connections between objects, components or controls.

**Other Types**

**TradValueOptions**

Describes how initialize target feature of the connection.

# TradConnection Object

Hierarchy Properties Methods

TradConnection is the common ancestor of all connection components.

**Unit**
radconlist

**Declaration**
```
TradConnection = class(TComponent)
```

**Description**
TradConnection provide visual connection between two objects. The descendant classes realize common connections: property to property, event to property and event to method. Do not create instances of TradConnection.

## TradConnection Properties

Properties  Methods

**In TradConnection**

- ConnectionList
- Index
- Source
- SourceInfo
- Target
- TargetInfo

## TradConnection Methods

Properties Methods

**In TradConnection**

- Connect
- Destroy
- GetParentComponent
- HasParent
- Unconnect

## ConnectionList property

Refers to the connection list that contains the connection.

**Applies to**
TradConnection

**Declaration**
```
Property ConnectionList : TradConnectionList;
```

**Description**
ConnectionList indicates which connection list component contains the connection. An connection list provides a visual interface for working with connections. The connection's properties are accessible in the Object Inspector by double-clicking on the connection list and selecting the connection.

# Index property

Holds the index of the connection in the connection list.

**Applies to**
TradConnection

**Declaration**

```
Property Index : Integer;
```

**Description**
Index contains the indexed value that indicates the location of the connection in the connection list. You can change the location of an connection in its list by using either the up and down arrows on either the toolbar or the speed menu of the connection list editor.

## Source property

Specifies the source object of the connection.

**Applies to**
TradConnection

**Declaration**

```
Property Source : TComponent;
```

**Description**
Source designates the source object that is associated with the connection.

**Note:** After set this property, the method Unconnect is called.

## SourceInfo property

Specifies the TrueRAD class info of Source object.

**Applies to**
TradConnection

**Declaration**

```
Property SourceInfo : TradClassInfo;
```

## Target property

Specifies the target object of the connection.

**Applies to**
TradConnection

**Declaration**

```
Property Target : TComponent;
```

**Description**
Target designates the target object that is associated with the connection.

**Note:** After set this property, the method Unconnect is called.

## TargetInfo property

Specifies the TrueRAD class info of Target object.

**Applies to**
TradConnection

**Declaration**

```
Property TargetInfo : TradClassInfo;
```

# Connect method

Initializes connection between two objects.

**Applies to**
TradConnection

**Declaration**

```
Function Connect: Boolean;
```

Virtual

Abstract

**Description**

The call of the given method initializes connection between two objects. If TradConnection owned by the connection list, the call of a method occurs automatically after loading the form from stream. The given method is abstract and the real operation is made in the descendants classes.

**Note:** At initialization of connection, the events of the object are intercepted by connection. If you create event handlers on design-time, all will be normal to work. If on run-time (after loading the form from stream), the connection can not work.

# Destroy method

Destroys the instance of the contained connection.

**Applies to**
TradConnection

**Declaration**

```
Procedure Destroy;
Override
```

**Description**
If the connection is contained in an connection list, Destroy removes the connection from that list. Then it calls the inherited Destroy.

# GetIndex method

**Applies to**
TradConnection

**Declaration**

```
Function GetIndex: Integer;
```

## GetParentComponent method

Returns the connection list that contains the connection.

**Applies to**
TradConnection

**Declaration**

```
Function GetParentComponent: TComponent;
Override
```

**Description**
GetParentComponent is called by the streaming system that loads and saves VCL components. It ensures that objects are loaded and saved with their parent objects. TradConnectionn overrides the inherited method to identify its associated connection list as its parent.

# GetSourceInfo method

**Applies to**
TradConnection

**Declaration**

```
Function GetSourceInfo: TradClassInfo;
```

# GetTargetInfo method

**Applies to**
TradConnection

**Declaration**

```
Function GetTargetInfo: TradClassInfo;
```

# HasParent method

Indicates whether the connection is contained in an connection list.

**Applies to**
TradConnection

**Declaration**
```
Function HasParent: Boolean;
```
Override

**Description**
HasParent is called by the streaming system that loads and saves VCL components. It ensures that objects are loaded and saved with their parent objects. TradConnection overrides the inherited method to return True if the connection is contained in an connection list.

# Notification method

**Applies to**
TradConnection

**Declaration**
```
Procedure Notification(AComponent: TComponent; Operation: TOperation);
Override
```

# ReadState method

**Applies to**
TradConnection

**Declaration**
```
Procedure ReadState(Reader: TReader);
Override
```

# SetConnectionList method

**Applies to**
TradConnection

**Declaration**

```
Procedure SetConnectionList(Value: TradCOnnectionList);
```

# SetIndex method

**Applies to**
TradConnection

**Declaration**

```
Procedure SetIndex(Value: Integer);
```

# SetParentComponent method

**Applies to**
TradConnection

**Declaration**
```
Procedure SetParentComponent(AParent: TComponent);
Override
```

# SetSource method

**Applies to**
TradConnection

**Declaration**

```
Procedure SetSource(Value: TComponent);
```

# SetTarget method

**Applies to**
TradConnection

**Declaration**

```
Procedure SetTarget(Value: TComponent);
```

# Unconnect method

Breaks off connection between two objects.

**Applies to**
TradConnection

**Declaration**

```
Procedure Unconnect;
```

Virtual

Abstract

**Description**
The call of the given method breaks off connection between two objects. The call of a method occurs automatically after changing Source or Target properties.

# TradConnectionList Component

Hierarchy Properties Methods

TradConnectionsList is the generic class for creating components that maintain connections between objects, components or controls.

**Unit**

radconlist

**Declaration**

```
TradConnectionList = class(TComponent)
```

**Description**

TradConnectionList is a component that allows you to work with connections at design-time. The connection list editor is the user interface for working with connections. From the connection list editor you can add, delete and modify connections.

## TradConnectionList Properties

Properties  Methods

**In TradConnectionList**

- ConnectionCount
  Connections

## TradConnectionList Methods

Properties  Methods

**In TradConnectionList**

- Create
- Destroy

# ConnectionCount property

Indicates the number of connections owned by the component.

**Applies to**
TradConnectionList

**Declaration**
```
Property ConnectionCount : Integer;
```

**Description**
Use ConnectionCount to find or verify the number of connections owned by a component, or when iterating through the Connections list to perform some action on all owned connections. ConnectionCount is used internally for such iterative procedures.

**Note:** The ConnectionCount of a component contains the same number of items as in the Connections list for that component, and is always 1 more than the highest Connections index, because the first Connections index is always 0.

## Connections property

Lists all connections owned by the component.

**Applies to**
TradConnectionList

**Declaration**
```
Property Connections[Index:Integer] : TradConnection;
```

**Description**
Use Connections to access any of the connections owned by this component. The Connections property is most useful when referring to owned connections by number rather than name. It is also used internally for iterative processing of all owned connections.

**Note:** For convenience use Connections with ConnectionCount for iterative processing. However, be aware that while the ConnectionCount of a component contains the same number of items as in the Connections list for that component, ConnectionCount is always 1 more than the highest Connections index, because the first Connections index is always 0.

# AddConnection method

**Applies to**
TradConnectionList

**Declaration**

```
Procedure AddConnection(Connection: TradConnection);
```

## Create method

Instantiates and initializes a TradConnectionList object.

**Applies to**
TradConnectionList

**Declaration**
```
Procedure Create(AOwner: TComponent);
Override
```

**Description**
When using the form designer, connection list instances are created automatically when you drop an connection list component on your form. The AOwner parameter is the owner of the action list, typically this is the form.
Create calls the inherited Create, and initializes the object by creating its sub-objects that handle the list of connections.

## Destroy method

Disposes of the instance of an connection list object.

**Applies to**
TradConnectionList

**Declaration**

```
Procedure Destroy;
Override
```

**Description**
You do not need to call the destructor for an connection list if it has an owner. The list is destroyed automatically by its owner that was passed to its constructor when it was created.

# GetChildren method

**Applies to**
TradConnectionList

**Declaration**
```
Procedure GetChildren(Proc: TGetChildProc; Root: TComponent);
Override
```

# GetConnection method

**Applies to**
TradConnectionList

**Declaration**

```
Function GetConnection(Index: Integer): TradConnection;
```

# GetConnectionCount method

**Applies to**
TradConnectionList

**Declaration**

```
Function GetConnectionCount: Integer;
```

# Loaded method

**Applies to**

TradConnectionList

**Declaration**

```
Procedure Loaded;
Override
```

# Notification method

**Applies to**
TradConnectionList

**Declaration**
```
Procedure Notification(AComponent: TComponent; Operation: TOperation);
Override
```

# RemoveConnection method

**Applies to**
TradConnectionList

**Declaration**
```
Procedure RemoveConnection(Connection: TradConnection);
```

# SetChildOrder method

**Applies to**
TradConnectionList

**Declaration**
```
Procedure SetChildOrder(Component: TComponent; Order: Integer);
Override
```

# SetConnection method

**Applies to**
TradConnectionList

**Declaration**

```
Procedure SetConnection(Index: Integer; Value: TradConnection);
```

## TradValueOptions type

Describes how initialize target feature of the connection.

**Unit**
radconlist

**Declaration**
```
TradValueOptions = (voConstant, voComponentProperty, voEventProperty);
```

**Description**

| | |
|---|---|
| voConstant | Set feature value as string constant. |
| voComponentProperty | Set feature value as component property. |
| voEventProperty | Set feature value as event parameter. |

**Hierarchy**

TComponent
|
TradConnection

**Direct subclasses**

TradE2MConnection
TradE2PConnection
TradP2PConnection

**Hierarchy**

TComponent
|
TradConnectionList

**Subclasses**
None

# rade2m Unit {button &Top,JI(`',`IDH_Unit_rade2m')}{button &Classes,JI(`',`IDH_UnitTopic_rade2m_Classes')}

Legend

**Classes**

TradE2MConnection
TradParameterConnection
TradParameterConnections

# TradE2MConnection Object

Hierarchy Properties Methods

**Unit**

rade2m

**Declaration**

```
TradE2MConnection = class(TradConnection)
```

## TradE2MConnection Properties

Properties  Methods

**In TradE2MConnection**

- SourceEvent
- SourceEventInfo
- TargetMethod
- TargetMethodInfo
- TargetMethodParameters

**Derived from TradConnection**

- ConnectionList
- Index
- Source
- SourceInfo
- Target
- TargetInfo

## TradE2MConnection Methods

Properties  Methods

**In TradE2MConnection**

- Connect
- Create
- Destroy
- Unconnect

**Derived from TradConnection**

- Connect
- Destroy
- GetParentComponent
- HasParent
- Unconnect

# SourceEvent property

**Applies to**
TradE2MConnection

**Declaration**

```
Property SourceEvent : String;
```

# SourceEventInfo property

**Applies to**
TradE2MConnection

**Declaration**

```
Property SourceEventInfo : TradEventInfo;
```

# TargetMethod property

**Applies to**
TradE2MConnection

**Declaration**

```
Property TargetMethod : String;
```

# TargetMethodInfo property

**Applies to**
TradE2MConnection

**Declaration**

```
Property TargetMethodInfo : TradMethodInfo;
```

# TargetMethodParameters property

**Applies to**
TradE2MConnection

**Declaration**
```
Property TargetMethodParameters : TradParameterConnections;
```

# Connect method

**Applies to**

TradE2MConnection

**Declaration**

```
Function Connect: Boolean;
Override
```

# Create method

**Applies to**

TradE2MConnection

**Declaration**

```
Procedure Create(AOwner: TComponent);
Override
```

# Destroy method

**Applies to**
TradE2MConnection

**Declaration**
```
Procedure Destroy;
Override
```

# GetSourceEventInfo method

**Applies to**
TradE2MConnection

**Declaration**

```
Function GetSourceEventInfo: TradEventInfo;
```

# GetTargetMethodInfo method

**Applies to**
TradE2MConnection

**Declaration**

```
Function GetTargetMethodInfo: TradMethodInfo;
```

# Notification method

**Applies to**
TradE2MConnection

**Declaration**
```
Procedure Notification(AComponent: TComponent; Operation: TOperation);
Override
```

# OnSourceEvent method

**Applies to**
TradE2MConnection

**Declaration**

```
Procedure OnSourceEvent(Sender: TObject; Event: TradEventInfo);
```

# SetTargetMethod method

**Applies to**
TradE2MConnection

**Declaration**

```
Procedure SetTargetMethod(Value: String);
```

# Unconnect method

**Applies to**

TradE2MConnection

**Declaration**

```
Procedure Unconnect;
Override
```

# TradParameterConnection Object

Hierarchy  Properties

**Unit**

rade2m

**Declaration**

```
TradParameterConnection = class(TCollectionItem)
```

## TradParameterConnection Properties

Properties

**In TradParameterConnection**

- Component
  - ComponentInfo
    - ComponentProperty
    - ComponentPropertyIndex
  - ComponentPropertyInfo
  - Connection
    - Constant
    - EventParameter
  - EventParameterInfo
    - Options
  - ParameterInfo
    - ParameterName

## Component property

**Applies to**
TradParameterConnection

**Declaration**
```
Property Component : TComponent;
```

# ComponentInfo property

**Applies to**
TradParameterConnection

**Declaration**
```
Property ComponentInfo : TradClassInfo;
```

## ComponentProperty property

**Applies to**
TradParameterConnection

**Declaration**
```
Property ComponentProperty : String;
```

# ComponentPropertyIndex property

**Applies to**
TradParameterConnection

**Declaration**

```
Property ComponentPropertyIndex : Variant;
```

# ComponentPropertyInfo property

**Applies to**
TradParameterConnection

**Declaration**
```
Property ComponentPropertyInfo : TradPropertyInfo;
```

# Connection property

**Applies to**
TradParameterConnection

**Declaration**

```
Property Connection : TradE2MConnection;
```

# Constant property

**Applies to**
TradParameterConnection

**Declaration**

```
Property Constant : String;
```

# EventParameter property

**Applies to**
TradParameterConnection

**Declaration**

```
Property EventParameter : String;
```

# EventParameterInfo property

**Applies to**
TradParameterConnection

**Declaration**

```
Property EventParameterInfo : TradParameterInfo;
```

# Options property

**Applies to**
TradParameterConnection

**Declaration**
```
Property Options : TradValueOptions;
```

# ParameterInfo property

**Applies to**
TradParameterConnection

**Declaration**

```
Property ParameterInfo : TradParameterInfo;
```

# ParameterName property

**Applies to**
TradParameterConnection

**Declaration**

```
Property ParameterName : String;
```

# DefineProperties method

**Applies to**
TradParameterConnection

**Declaration**
```
Procedure DefineProperties(Filer: TFiler);
Override
```

# GetComponentInfo method

**Applies to**
TradParameterConnection

**Declaration**

```
Function GetComponentInfo: TradClassInfo;
```

# GetComponentPropertyInfo method

**Applies to**
TradParameterConnection

**Declaration**

```
Function GetComponentPropertyInfo: TradPropertyInfo;
```

# GetConnection method

**Applies to**
TradParameterConnection

**Declaration**

```
Function GetConnection: TradE2MConnection;
```

# GetDisplayName method

**Applies to**
TradParameterConnection

**Declaration**

```
Function GetDisplayName: string;
```

Override

# GetEventParameterInfo method

**Applies to**
TradParameterConnection

**Declaration**

```
Function GetEventParameterInfo: TradParameterInfo;
```

# GetParameterInfo method

**Applies to**
TradParameterConnection

**Declaration**

```
Function GetParameterInfo: TradParameterInfo;
```

# ReadComponentPropertyIndexes method

**Applies to**
TradParameterConnection

**Declaration**

```
Procedure ReadComponentPropertyIndexes(Reader: TReader);
```

# SetComponent method

**Applies to**
TradParameterConnection

**Declaration**

```
Procedure SetComponent(Value: TComponent);
```

# SetComponentProperty method

**Applies to**
TradParameterConnection

**Declaration**

```
Procedure SetComponentProperty(Value: String);
```

# WriteComponentPropertyIndexes method

**Applies to**
TradParameterConnection

**Declaration**

```
Procedure WriteComponentPropertyIndexes(Writer: TWriter);
```

# TradParameterConnections Object

Hierarchy Properties Methods

**Unit**

rade2m

**Declaration**

```
TradParameterConnections = class(TCollection)
```

## TradParameterConnections Properties

Properties  Methods

**In TradParameterConnections**

Override Connection

Items

Method

## TradParameterConnections Methods

Properties  Methods

**In TradParameterConnections**

- Add
- Create
- Init

# Connection property

**Applies to**
TradParameterConnections

**Declaration**

```
Property Connection : TradE2MConnection;
```

# Items property

**Applies to**
TradParameterConnections

**Declaration**
```
Property Items[Index:Integer] : TradParameterConnection;
```

# Method property

**Applies to**
TradParameterConnections

**Declaration**

```
Property Method : String;
```

# Add method

**Applies to**
TradParameterConnections

**Declaration**
```
Function Add: TradParameterConnection;
```

# Create method

**Applies to**
TradParameterConnections

**Declaration**

```
Procedure Create(AConnector: TradE2MConnection);
```

# GetLink method

**Applies to**
TradParameterConnections

**Declaration**

```
Function GetLink(Index: Integer): TradParameterConnection;
```

# GetOwner method

**Applies to**
TradParameterConnections

**Declaration**
```
Function GetOwner: TPersistent;
Override
```

# Init method

**Applies to**
TradParameterConnections

**Declaration**

```
Procedure Init(Method: TradMethodInfo);
```

# SetLink method

**Applies to**
TradParameterConnections

**Declaration**

```
Procedure SetLink(Index: Integer; Value: TradParameterConnection);
```

**Hierarchy**

TComponent
|
TradConnection
|
TradE2MConnection

**Subclasses**

None

**Hierarchy**

TCollectionItem
|
TradParameterConnection

**Subclasses**
None

**Hierarchy**

TCollection
|
TradParameterConnections

**Subclasses**
None

# rade2p Unit {button &Top,JI(`',`IDH_Unit_rade2p')}{button &Classes,JI(`',`IDH_UnitTopic_rade2p_Classes')}

Legend

**Classes**

TradE2PConnection

# TradE2PConnection Object

Hierarchy Properties Methods

**Unit**

rade2p

**Declaration**

```
TradE2PConnection = class(TradConnection)
```

## TradE2PConnection Properties

Properties  Methods

**In TradE2PConnection**

Override SourceEvent

Override SourceEventInfo

Override TargetProperty

Override TargetPropertyIndex

Override TargetPropertyInfo

Override ValueComponent

Override ValueComponentInfo

Override ValueComponentProperty

Override ValueComponentPropertyIndex

Override ValueComponentPropertyInfo

Override ValueConstant

Override ValueEventParameter

Override ValueEventParameterInfo

Override ValueOptions

**Derived from TradConnection**

ConnectionList
Index

Override Source

Override SourceInfo

Override Target

Override TargetInfo

## TradE2PConnection Methods

Properties  Methods

**In TradE2PConnection**

- Connect
- Create
- Destroy
- Unconnect

**Derived from TradConnection**

- Connect
- Destroy
- GetParentComponent
- HasParent
- Unconnect

# SourceEvent property

**Applies to**
TradE2PConnection

**Declaration**

```
Property SourceEvent : String;
```

# SourceEventInfo property

**Applies to**
TradE2PConnection

**Declaration**

```
Property SourceEventInfo : TradEventInfo;
```

## TargetProperty property

**Applies to**
TradE2PConnection

**Declaration**

```
Property TargetProperty : String;
```

# TargetPropertyIndex property

**Applies to**
TradE2PConnection

**Declaration**

```
Property TargetPropertyIndex : Variant;
```

# TargetPropertyInfo property

**Applies to**
TradE2PConnection

**Declaration**

```
Property TargetPropertyInfo : TradPropertyInfo;
```

# ValueComponent property

**Applies to**
TradE2PConnection

**Declaration**
```
Property ValueComponent : TComponent;
```

# ValueComponentInfo property

**Applies to**
TradE2PConnection

**Declaration**
```
Property ValueComponentInfo : TradClassInfo;
```

## ValueComponentProperty property

**Applies to**
TradE2PConnection

**Declaration**
```
Property ValueComponentProperty : String;
```

# ValueComponentPropertyIndex property

**Applies to**
TradE2PConnection

**Declaration**

```
Property ValueComponentPropertyIndex : Variant;
```

# ValueComponentPropertyInfo property

**Applies to**
TradE2PConnection

**Declaration**

```
Property ValueComponentPropertyInfo : TradPropertyInfo;
```

# ValueConstant property

**Applies to**
TradE2PConnection

**Declaration**

```
Property ValueConstant : String;
```

# ValueEventParameter property

**Applies to**
TradE2PConnection

**Declaration**

```
Property ValueEventParameter : String;
```

# ValueEventParameterInfo property

**Applies to**
TradE2PConnection

**Declaration**
```
Property ValueEventParameterInfo : TradParameterInfo;
```

# ValueOptions property

**Applies to**
TradE2PConnection

**Declaration**
```
Property ValueOptions : TradValueOptions;
```

# Connect method

**Applies to**

TradE2PConnection

**Declaration**

```
Function Connect: Boolean;
Override
```

# Create method

**Applies to**

TradE2PConnection

**Declaration**

```
Procedure Create(AOwner: TComponent);
Override
```

# DefineProperties method

**Applies to**
TradE2PConnection

**Declaration**
```
Procedure DefineProperties(Filer: TFiler);
Override
```

# Destroy method

**Applies to**

TradE2PConnection

**Declaration**

```
Procedure Destroy;
Override
```

# GetSourceEventInfo method

**Applies to**
TradE2PConnection

**Declaration**

```
Function GetSourceEventInfo: TradEventInfo;
```

# GetTargetPropertyInfo method

**Applies to**
TradE2PConnection

**Declaration**

```
Function GetTargetPropertyInfo: TradPropertyInfo;
```

# GetValueComponentInfo method

**Applies to**
TradE2PConnection

**Declaration**

```
Function GetValueComponentInfo: TradClassInfo;
```

# GetValueComponentPropertyInfo method

**Applies to**
TradE2PConnection

**Declaration**

```
Function GetValueComponentPropertyInfo: TradPropertyInfo;
```

# GetValueEventParameterInfo method

**Applies to**
TradE2PConnection

**Declaration**

```
Function GetValueEventParameterInfo: TradParameterInfo;
```

# Notification method

**Applies to**
TradE2PConnection

**Declaration**
```
Procedure Notification(AComponent: TComponent; Operation: TOperation);
Override
```

# OnSourceEvent method

**Applies to**
TradE2PConnection

**Declaration**

```
Procedure OnSourceEvent(Sender: TObject; Event: TradEventInfo);
```

# ReadTargetPropertyIndexes method

**Applies to**
TradE2PConnection

**Declaration**

```
Procedure ReadTargetPropertyIndexes(Reader: TReader);
```

# ReadValueComponentPropertyIndexes method

**Applies to**
TradE2PConnection

**Declaration**

```
Procedure ReadValueComponentPropertyIndexes(Reader: TReader);
```

# SetTargetProperty method

**Applies to**
TradE2PConnection

**Declaration**

```
Procedure SetTargetProperty(Value: String);
```

# SetValueComponent method

**Applies to**
TradE2PConnection

**Declaration**

```
Procedure SetValueComponent(Value: TComponent);
```

# SetValueComponentProperty method

**Applies to**
TradE2PConnection

**Declaration**

```
Procedure SetValueComponentProperty(Value: String);
```

# Unconnect method

**Applies to**

TradE2PConnection

**Declaration**

```
Procedure Unconnect;
Override
```

# WriteTargetPropertyIndexes method

**Applies to**
TradE2PConnection

**Declaration**

```
Procedure WriteTargetPropertyIndexes(Writer: TWriter);
```

# WriteValueComponentPropertyIndexes method

**Applies to**
TradE2PConnection

**Declaration**

```
Procedure WriteValueComponentPropertyIndexes(Writer: TWriter);
```

**Hierarchy**

TComponent
|
TradConnection
|
TradE2PConnection

**Subclasses**

None

# radp2p Unit {button &Top,JI(`',`IDH_Unit_radp2p')}{button &Classes,JI(`',`IDH_UnitTopic_radp2p_Classes')}

Legend

**Classes**

TradP2PConnection

TradP2PConnection is the components which realizes 'property to property' connection.

# TradP2PConnection Object

Hierarchy Properties Methods

TradP2PConnection is the components which realizes 'property to property' connection.

**Unit**

radp2p

**Declaration**

```
TradP2PConnection = class(TradConnection)
```

## TradP2PConnection Properties

Properties  Methods

**In TradP2PConnection**

Override SourceEvent

Override SourceEventInfo

Override SourceProperty

Override SourcePropertyIndex

Override SourcePropertyInfo

Override TargetEvent

Override TargetEventInfo

Override TargetProperty

Override TargetPropertyIndex

Override TargetPropertyInfo

**Derived from TradConnection**

ConnectionList Index

Override Source

Override SourceInfo

Override Target

Override TargetInfo

## TradP2PConnection Methods

Properties  Methods

**In TradP2PConnection**

- Connect
- Create
- Destroy
- Unconnect

**Derived from TradConnection**

- Connect
- Destroy
- GetParentComponent
- HasParent
- Unconnect

# SourceEvent property

**Applies to**
TradP2PConnection

**Declaration**

```
Property SourceEvent : String;
```

# SourceEventInfo property

**Applies to**
TradP2PConnection

**Declaration**

```
Property SourceEventInfo : TradEventInfo;
```

## SourceProperty property

**Applies to**
TradP2PConnection

**Declaration**
```
Property SourceProperty : String;
```

# SourcePropertyIndex property

**Applies to**
TradP2PConnection

**Declaration**

```
Property SourcePropertyIndex : Variant;
```

## SourcePropertyInfo property

**Applies to**
TradP2PConnection

**Declaration**

```
Property SourcePropertyInfo : TradPropertyInfo;
```

# TargetEvent property

**Applies to**
TradP2PConnection

**Declaration**

```
Property TargetEvent : String;
```

# TargetEventInfo property

**Applies to**
TradP2PConnection

**Declaration**

```
Property TargetEventInfo : TradEventInfo;
```

## TargetProperty property

**Applies to**
TradP2PConnection

**Declaration**

```
Property TargetProperty : String;
```

# TargetPropertyIndex property

**Applies to**
TradP2PConnection

**Declaration**

```
Property TargetPropertyIndex : Variant;
```

# TargetPropertyInfo property

**Applies to**
TradP2PConnection

**Declaration**

```
Property TargetPropertyInfo : TradPropertyInfo;
```

# Connect method

**Applies to**

TradP2PConnection

**Declaration**

```
Function Connect: Boolean;
Override
```

# Create method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure Create(AOwner: TComponent);
Override
```

# DefineProperties method

**Applies to**
TradP2PConnection

**Declaration**
```
Procedure DefineProperties(Filer: TFiler);
Override
```

# Destroy method

**Applies to**
TradP2PConnection

**Declaration**
```
Procedure Destroy;
Override
```

# GetSourceEventInfo method

**Applies to**
TradP2PConnection

**Declaration**

```
Function GetSourceEventInfo: TradEventInfo;
```

# GetSourcePropertyInfo method

**Applies to**
TradP2PConnection

**Declaration**

```
Function GetSourcePropertyInfo: TradPropertyInfo;
```

# GetTargetEventInfo method

**Applies to**
TradP2PConnection

**Declaration**

```
Function GetTargetEventInfo: TradEventInfo;
```

# GetTargetPropertyInfo method

**Applies to**
TradP2PConnection

**Declaration**

```
Function GetTargetPropertyInfo: TradPropertyInfo;
```

# ReadSourcePropertyIndexes method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure ReadSourcePropertyIndexes(Reader: TReader);
```

# ReadTargetPropertyIndexes method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure ReadTargetPropertyIndexes(Reader: TReader);
```

# SetSourceProperty method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure SetSourceProperty(Value: String);
```

# SetSourceValue method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure SetSourceValue;
```

# SetTargetProperty method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure SetTargetProperty(Value: String);
```

# SetTargetValue method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure SetTargetValue;
```

# SourcePropertyChange method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure SourcePropertyChange(Sender: TObject; Event: TradEventInfo);
```

# TargetPropertyChange method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure TargetPropertyChange(Sender: TObject; Event: TradEventInfo);
```

# Unconnect method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure Unconnect;
```

Override

# WriteSourcePropertyIndexes method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure WriteSourcePropertyIndexes(Writer: TWriter);
```

# WriteTargetPropertyIndexes method

**Applies to**
TradP2PConnection

**Declaration**

```
Procedure WriteTargetPropertyIndexes(Writer: TWriter);
```

**Hierarchy**

TComponent
|
TradConnection
|
TradP2PConnection

**Subclasses**

None

# radvar Unit {button &Top,JI(`',`IDH_Unit_radvar')}{button &Classes,JI(`',`IDH_UnitTopic_radvar_Classes')}

Legend

**Classes**

TradVariable

TradVariableClassInfo

# Override TradVariable Component

Hierarchy Properties Methods

**Unit**

radvar

**Declaration**

```
TradVariable = class(TComponent, IradObject)
```

## TradVariable Properties

Properties Methods

**In TradVariable**

Override ClassTypeName

## TradVariable Methods

Properties  Methods

**In TradVariable**

- Create
- Destroy
- GetRADClassInfo

# ClassTypeName property

**Applies to**
TradVariable

**Declaration**
```
Property ClassTypeName : String;
```

# Create method

**Applies to**
TradVariable

**Declaration**
```
Procedure Create(AOwner: TComponent);
Override
```

# Destroy method

**Applies to**

TradVariable

**Declaration**

```
Procedure Destroy;
Override
```

# GetRADClassInfo method

**Applies to**
TradVariable

**Declaration**

```
Function GetRADClassInfo: TradClassInfo;
```

# SetClassTypeName method

**Applies to**
TradVariable

**Declaration**

```
Procedure SetClassTypeName(Value: String);
```

# TradVariableClassInfo Object

Hierarchy  Methods

**Unit**

radvar

**Declaration**

```
TradVariableClassInfo = class(TradClassInfo)
```

## TradVariableClassInfo Methods

Methods

**In TradVariableClassInfo**

- Create
- Destroy
- GetProperty
- InvokeMethod
- SetProperty

# Create method

**Applies to**
TradVariableClassInfo

**Declaration**

```
Procedure Create(ClassTypeName: String);
```

# DefineRADEvents method

**Applies to**
TradVariableClassInfo

**Declaration**
```
Procedure DefineRADEvents(AEvents: TList);
Override
```

# DefineRADMethods method

**Applies to**
TradVariableClassInfo

**Declaration**
```
Procedure DefineRADMethods(AMethods: TList);
Override
```

# DefineRADProperties method

**Applies to**
TradVariableClassInfo

**Declaration**
```
Procedure DefineRADProperties(AProperties: TList);
Override
```

# Destroy method

**Applies to**

TradVariableClassInfo

**Declaration**

```
Procedure Destroy;
Override
```

# GetProperty method

**Applies to**
TradVariableClassInfo

**Declaration**
```
Function GetProperty(Prop: TradPropertyInfo): Variant;
Override
```

# InvokeMethod method

**Applies to**
TradVariableClassInfo

**Declaration**

```
Procedure InvokeMethod(Method: TradMethodInfo);
Override
```

# SetProperty method

**Applies to**
TradVariableClassInfo

**Declaration**

```
Procedure SetProperty(Prop: TradPropertyInfo; Value: Variant);
Override
```

# Update method

**Applies to**
TradVariableClassInfo

**Declaration**

```
Procedure Update(ClassTypeName: String);
```

**Hierarchy**

TComponent
|
TradVariable

**Subclasses**
None

**Hierarchy**

TradClassInfo
|
TradVariableClassInfo

**Subclasses**

None

**Welcome to TrueRAD Suite!**

We sincerely hope (and think) that you will like this quality product!

**TrueRAD Suite** is a set of components for visual programming. This package will allow you to create applications without writing any code! For obtaining more detailed information look the following topics:

**Overview/Introduction**

TrueRAD Suite is a set of components for visual programming. This package will allow you to create applications without writing any code. The given possibility is grounded on binding of objects, components, and controls among themselves by connectors. At present there are four main connectors: property-to-property, event-to-property, event-to-method and parameter-from-feature.

For obtaining more detailed information look the following topics:

Use Connections - How to use connections.

### Use connections

When you make a connection in the connection editor, you define the interaction between components. For example, if you want a data value to change when an event occurs, you would make an event-to-property connection. The following table summarizes the types of connections that the connection editor provides.

| If you want to... | Use this connection type | Connection class |
|---|---|---|
| Cause one data value to change another | property-to-property | TradP2PConnection |
| Change a data value whenever an event occurs | event-to-property | TradE2PConnection |
| Call a public behavior whenever an event occurs | event-to-method | TradE2MConnection |
| Supply a value to a parameter | parameter-from-feature | TradParameterConnection |

## Licence, copyright and disclaimer

### Copyright notice



### Liability disclaimer

**How to order**

At present TrueRAD Suite it is distributed free-of-charge for noncommercial usage. If you want to order TrueRAD Suite, send the mail with a 'order' subject to truerad2000@yahoo.com.

---

{button ,AL("ContactAndProductInfo")} Related Topics

**Technical Support**

TrueRAD Soft offers two technical support: via email and the internet.

**Via email**

To obtain technical support for TrueRAD Suite, please send an electronic mail message to truerad2000@yahoo.com. Be sure to include a detailed description of the problem you are having in addition to the following information:

- Operating System and Version Number
- Version number of development tool (i.e.Delphi or C++Builder)

**On the Web**

Be sure to visit our Web site for product updates, tips and techniques, and upgrade information. Point your browser to www.geocities.com/truerad_2000.

---

{button ,AL("ContactAndProductInfo")} Related Topics