

IPS7RdMulti Members

Namespace: IPS7Lnk

Assemblies: IPS7LnkNet.Advanced.dll, IPS7LnkNet.Advanced.dll

The [IPS7RdMulti](#) type exposes the following members.

Constructors

IPS7RdMulti()

Constructor

C#

```
public IPS7RdMulti()
```

Properties

Cnt

Anzahl der Elemente des Types [DataType](#)

C#

```
public int Cnt { get; set; }
```

Property Value

[Int32](#)

DataArea

Datenbereich in der SPS, siehe: [DA](#)

C#

```
public int DataArea { get; set; }
```

Property Value

[Int32](#)

DataType

Datentyp in der SPS, , siehe: [S7DataTypes](#)

C#

```
public int DataType { get; set; }
```

Property Value

Int32

DBNr

Nummer des Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

C#

```
public int DBNr { get; set; }
```

Property Value

Int32

ErrMemSize

MemSize Fehler, -20, wenn user zu kleines Array übergibt/>

C#

```
public int ErrMemSize { get; set; }
```

Property Value

Int32

PcDataType

Hier wird der Datentyp im PC abgelegt.

C#

```
public int PcDataType { get; set; }
```

Property Value

Int32

Result

Nach erfolgreichem Aufruf, wird hier das Ergebnis für diesen Eintrag abgelegt siehe: [Result](#)

C#

```
public int Result { get; set; }
```

Property Value

Int32

Start

Nummer des ersten Elements **DataType** in der SPS

C#

```
public int Start { get; set; }
```

Property Value

Int32

StartBit

Nummer des ersten Bits in der SPS (wird nur bei Bitfunktionen ausgewertet)

C#

```
public int StartBit { get; set; }
```

Property Value

Int32

UserData0

UserData0 und **Userdata1** sind Elemente, die der Aufrufer frei verwenden kann, um z.B. Eigenschaften für diesen Eintrag zu kennzeichnen.

C#

```
public int UserData0 { get; set; }
```

Property Value

Int32

UserData1

UserData0 und **Userdata1** sind Elemente, die der Aufrufer frei verwenden kann, um z.B. Eigenschaften für diesen Eintrag zu kennzeichnen.

C#

```
public int UserData1 { get; set; }
```

Property Value

Int32

Methods

Bit(Char, Int32, Int32, Int32, Int32, Byte[])

Bit-Zugriff, Datentyp der Variable im PC: byte[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, byte[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Byte[]

Returns

Int32

Remarks

Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

Bit(Char, Int32, Int32, Int32, Int32, Char[])

Bit-Zugriff, Datentyp der Variable im PC: sbyte[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, char[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Char[]

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, Double[])

Bit-Zugriff, Datentyp der Variable im PC: double[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Double[]

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, Int16[])

Bit-Zugriff, Datentyp der Variable im PC: short[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, short[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Int16[]

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, Int32[])

Bit-Zugriff, Datentyp der Variable im PC: int[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, int[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Int32[]

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, ref Byte)

Bit-Zugriff: Datentyp der Variable im PC: ref byte

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref byte Buf)
```

Parameters

DataArea Char

Datenbereich, siehe: DA

DBNr Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

StartBit Int32

Nummer des ersten Bits, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Byte

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Bit(Char, Int32, Int32, Int32, Int32, ref Char)

Bit-Zugriff, Datentyp der Variable im PC: ref char Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref char Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Char

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, ref Double)

Bit-Zugriff, Datentyp der Variable im PC: ref double Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref double Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Double

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, ref Int16)

Bit-Zugriff, Datentyp der Variable im PC: ref short Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref short Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Int16

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, ref Int32)

Bit-Zugriff, Datentyp der Variable im PC: ref int Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref int Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Int32

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, ref SByte)

Bit-Zugriff, Datentyp der Variable im PC: ref sbyte Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref sbyte Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf SByte

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, ref Single)

Bit-Zugriff, Datentyp der Variable im PC: ref float Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref float Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Single

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, ref UInt16)

Bit-Zugriff, Datentyp der Variable im PC: ref ushort Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref ushort Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf UInt16

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, ref UInt32)

Bit-Zugriff, Datentyp der Variable im PC: ref uint Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ref uint Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf UInt32

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, SByte[])

Bit-Zugriff, Datentyp der Variable im PC: sbyte[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, sbyte[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf SByte[]

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, Single[])

Bit-Zugriff, Datentyp der Variable im PC: float[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf Single[]

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, UInt16[])

Bit-Zugriff, Datentyp der Variable im PC: ushort[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, ushort[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf UInt16[]

Returns

Int32

Bit(Char, Int32, Int32, Int32, Int32, UInt32[])

Bit-Zugriff, Datentyp der Variable im PC: uint[] Beschreibung: Bit(Char, Int32, Int32, Int32, Int32, Byte%40)

C#

```
public int Bit(char DataArea, int DBNr, int Start, int StartBit, int Cnt, uint[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

StartBit Int32

Cnt Int32

Buf UInt32[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, Byte[])

Byte-Zugriff, Datentyp der Variable im PC: byte[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, byte[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Byte[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, Char[])

Byte-Zugriff, Datentyp der Variable im PC: char[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, char[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Char[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, Double[])

Byte-Zugriff, Datentyp der Variable im PC: double[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, Int16[])

Byte-Zugriff, Datentyp der Variable im PC: short[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, short[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int16[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, Int32[])

Byte-Zugriff, Datentyp der Variable im PC: int[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, int[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, ref Byte)

Byte-Zugriff: Datentyp der Variable im PC: ref byte

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref byte Buf)
```

Parameters

DataArea Char

Datenbereich, siehe: DA

DBNr Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Byte

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Byte(Char, Int32, Int32, Int32, ref Char)

Byte-Zugriff, Datentyp der Variable im PC: char Beschreibung: `Byte(Char, Int32, Int32, Int32, Byte%40)`

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref char Buf)
```

Parameters

`DataArea` Char

`DBNr` Int32

`Start` Int32

`Cnt` Int32

`Buf` Char

Returns

Int32

Byte(Char, Int32, Int32, Int32, ref Double)

Byte-Zugriff, Datentyp der Variable im PC: double Beschreibung: `Byte(Char, Int32, Int32, Int32, Byte%40)`

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

`DataArea` Char

`DBNr` Int32

`Start` Int32

Cnt Int32

Buf Double

Returns

Int32

Byte(Char, Int32, Int32, Int32, ref Int16)

Byte-Zugriff, Datentyp der Variable im PC: short Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref short Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int16

Returns

Int32

Byte(Char, Int32, Int32, Int32, ref Int32)

Byte-Zugriff, Datentyp der Variable im PC: int Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref int Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32

Returns

Int32

Byte(Char, Int32, Int32, Int32, ref SByte)

Byte-Zugriff, Datentyp der Variable im PC: sbyte Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref sbyte Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf SByte

Returns

Int32

Byte(Char, Int32, Int32, Int32, ref Single)

Byte-Zugriff, Datentyp der Variable im PC: float Beschreibung: `Byte(Char, Int32, Int32, Int32, Byte%40)`

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters

`DataArea` Char

`DBNr` Int32

`Start` Int32

`Cnt` Int32

`Buf` Single

Returns

Int32

Byte(Char, Int32, Int32, Int32, ref UInt16)

Byte-Zugriff, Datentyp der Variable im PC: ushort Beschreibung: `Byte(Char, Int32, Int32, Int32, Byte%40)`

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref ushort Buf)
```

Parameters

`DataArea` Char

`DBNr` Int32

`Start` Int32

Cnt Int32

Buf UInt16

Returns

Int32

Byte(Char, Int32, Int32, Int32, ref UInt32)

Byte-Zugriff, Datentyp der Variable im PC: uint Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ref uint Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32

Returns

Int32

Byte(Char, Int32, Int32, Int32, SByte[])

Byte-Zugriff, Datentyp der Variable im PC: sbyte[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, sbyte[] Buf)
```


Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf SByte[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, Single[])

Byte-Zugriff, Datentyp der Variable im PC: float[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, UInt16[])

Byte-Zugriff, Datentyp der Variable im PC: ushort[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, ushort[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt16[]

Returns

Int32

Byte(Char, Int32, Int32, Int32, UInt32[])

Byte-Zugriff, Datentyp der Variable im PC: uint[] Beschreibung: Byte(Char, Int32, Int32, Int32, Byte%40)

C#

```
public int Byte(char DataArea, int DBNr, int Start, int Cnt, uint[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32[]

Returns

Int32

Counter(Char, Int32, Int32, Int32, Char[])

Zähler/ Counter Datentyp der Variable im PC: char [] Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, char[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Char[]

Returns

Int32

Counter(Char, Int32, Int32, Int32, Double[])

Zähler/ Counter Datentyp der Variable im PC: double [] Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

Counter(Char, Int32, Int32, Int32, Int16[])

Zähler/ Counter Datentyp der Variable im PC: short [] Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, short[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int16[]

Returns

Int32

Counter(Char, Int32, Int32, Int32, Int32[])

Zähler/ Counter Datentyp der Variable im PC: int [] Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, int[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32[]

Returns

Int32

Counter(Char, Int32, Int32, Int32, ref Char)

Zähler/ Counter Datentyp der Variable im PC: char Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, ref char Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Char

Returns

Int32

Counter(Char, Int32, Int32, Int32, ref Double)

Zähler/ Counter Datentyp der Variable im PC: double Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double

Returns

Int32

Counter(Char, Int32, Int32, Int32, ref Int16)

Zähler/Counter Datentyp der Variable im PC: short

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, ref short Buf)
```

Parameters

DataArea Char

Datenbereich, siehe: [DA](#)

DBNr Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Int16

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Counter(Char, Int32, Int32, Int32, ref Int32)

Zähler/ Counter Datentyp der Variable im PC: int Beschreibung: [Counter\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, ref int Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32

Returns

Int32

Counter(Char, Int32, Int32, Int32, ref Single)

Zähler/ Counter Datentyp der Variable im PC: float Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single

Returns

Int32

Counter(Char, Int32, Int32, Int32, ref UInt16)

Zähler/ Counter Datentyp der Variable im PC: ushort [] Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, ref ushort Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt16

Returns

Int32

Counter(Char, Int32, Int32, Int32, ref UInt32)

Zähler/ Counter Datentyp der Variable im PC: uint Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, ref uint Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32

Returns

Int32

Counter(Char, Int32, Int32, Int32, Single[])

Zähler/ Counter Datentyp der Variable im PC: float [] Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

Counter(Char, Int32, Int32, Int32, UInt16[])

Zähler/ Counter Datentyp der Variable im PC: ushort [] Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, ushort[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt16[]

Returns

Int32

Counter(Char, Int32, Int32, Int32, UInt32[])

Zähler/ Counter Datentyp der Variable im PC: uint [] Beschreibung: Counter(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Counter(char DataArea, int DBNr, int Start, int Cnt, uint[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32[]

Returns

Int32

DWord(Char, Int32, Int32, Int32, Double[])

DWord (32 Bit)-Zugriff: Datentyp der Variable im PC: double[] Beschreibung: DWord(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int DWord(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

DWord(Char, Int32, Int32, Int32, Int32[])

DWord (32 Bit)-Zugriff: Datentyp der Variable im PC: int[] Beschreibung: DWord(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int DWord(char DataArea, int DBNr, int Start, int Cnt, int[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32[]

Returns

Int32

DWord(Char, Int32, Int32, Int32, ref Double)

DWord (32 Bit)-Zugriff: Datentyp der Variable im PC: double Beschreibung: [DWord\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int DWord(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

[DataArea](#) Char

[DBNr](#) Int32

[Start](#) Int32

[Cnt](#) Int32

[Buf](#) Double

Returns

Int32

DWord(Char, Int32, Int32, Int32, ref Int32)

DWord (32 Bit)-Zugriff: Datentyp der Variable im PC: int

C#

```
public int DWord(char DataArea, int DBNr, int Start, int Cnt, ref int Buf)
```

Parameters

[DataArea](#) Char

Datenbereich, siehe: [DA](#)

[DBNr](#) Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Int32

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

DWord(Char, Int32, Int32, Int32, ref Single)

DWord (32 Bit)-Zugriff: Datentyp der Variable im PC: float Beschreibung: DWord(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int DWord(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters**DataArea** Char**DBNr** Int32**Start** Int32**Cnt** Int32**Buf** Single**Returns**

Int32

DWord(Char, Int32, Int32, Int32, ref UInt32)

DWord (32 Bit)-Zugriff: Datentyp der Variable im PC: uint Beschreibung: DWord(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int DWord(char DataArea, int DBNr, int Start, int Cnt, ref uint Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32

Returns

Int32

DWord(Char, Int32, Int32, Int32, Single[])

DWord (32 Bit)-Zugriff: Datentyp der Variable im PC: float[] Beschreibung: DWord(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int DWord(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

DWord(Char, Int32, Int32, Int32, UInt32[])

DWord (32 Bit)-Zugriff: Datentyp der Variable im PC: uint[] Beschreibung: DWord(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int DWord(char DataArea, int DBNr, int Start, int Cnt, uint[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32[]

Returns

Int32

GetData(Byte[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(byte[] Buf)
```


Parameters

Buf Byte[]

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(Char[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(char[] Buf)
```

Parameters

Buf Char[]

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(Double[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(double[] Buf)
```

Parameters

Buf Double[]

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(Int16[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(short[] Buf)
```

Parameters

Buf [Int16\[\]](#)

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns[Int32](#)

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(Int32[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
[Obfuscation]  
public int GetData(int[] Buf)
```

Parameters

Buf [Int32\[\]](#)

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns[Int32](#)

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(Int64[])

C#

```
public int GetData(long[] Buf)
```

Parameters

Buf Int64[]

Returns

Int32

GetData(ref Byte)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref byte Buf)
```

Parameters

Buf Byte

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref Char)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref char Buf)
```

Parameters

Buf Char

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref Double)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref double Buf)
```

Parameters

Buf Double

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref Int16)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref short Buf)
```

Parameters

Buf Int16

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref Int32)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
[Obfuscation]  
public int GetData(ref int Buf)
```

Parameters

Buf [Int32](#)

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

[Int32](#)

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref Int64)

C#

```
public int GetData(ref long Buf)
```

Parameters

Buf [Int64](#)

Returns

[Int32](#)

GetData(ref SByte)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref sbyte Buf)
```

Parameters

Buf [SByte](#)

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

[Int32](#)

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref Single)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wie vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref float Buf)
```

Parameters

[Buf](#) [Single](#)

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

[Int32](#)

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref String)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wie vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref string Buf)
```

Parameters

[Buf](#) [String](#)

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

[Int32](#)

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref UInt16)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref ushort Buf)
```

Parameters

Buf [UInt16](#)

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

[Int32](#)

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref UInt32)

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ref uint Buf)
```

Parameters

Buf [UInt32](#)

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

[Int32](#)

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(ref UInt64)

C#

```
public int GetData(ref ulong Buf)
```

Parameters

Buf UInt64

Returns

Int32

GetData(SByte[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(sbyte[] Buf)
```

Parameters

Buf SByte[]

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(Single[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(float[] Buf)
```

Parameters

Buf Single[]

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(UInt16[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(ushort[] Buf)
```

Parameters

Buf UInt16[]

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(UInt32[])

Liest die Daten vorhergehenden RdMultiBuffered aus dem Cache, ACHTUNG! Es wird die gesamte Länge des Datenblocks gelesen, wir vorher mit den Methoden Bit, Byte, Word etc. angefordert wurde.

C#

```
public int GetData(uint[] Buf)
```

Parameters

Buf UInt32[]

ist der Zielspeicher in der Applikation. Als Parameter Buf unbedingt dieselbe Variable wie bei der Anforderung (Bit, Byte etc.) verwenden

Returns

Int32

Liefert als Returnwert das Ergebnis des letzten RdMultiBuffered-Aufrufs für diese Anforderung siehe: [Result](#)

GetData(UInt64[])

C#

```
public int GetData(ulong[] Buf)
```

Parameters

Buf UInt64[]

Returns

Int32

Int16(Char, Int32, Int32, Int32, Char[])

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: char[] Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, char[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Char[]

Returns

Int32

Int16(Char, Int32, Int32, Int32, Double[])

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: double[] Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

Int16(Char, Int32, Int32, Int32, Int16[])

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: short[] Beschreibung: Int16(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, short[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int16[]

Returns

Int32

Int16(Char, Int32, Int32, Int32, Int32[])

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: int[] Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, int[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32[]

Returns

Int32

Int16(Char, Int32, Int32, Int32, ref Char)

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: char Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, ref char Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Char

Returns

Int32

Int16(Char, Int32, Int32, Int32, ref Double)

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: double Beschreibung: Int16(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double

Returns

Int32

Int16(Char, Int32, Int32, Int32, ref Int16)

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: short

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, ref short Buf)
```

Parameters

DataArea Char

Datenbereich, siehe: [DA](#)

DBNr Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Int16

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Int16(Char, Int32, Int32, Int32, ref Int32)

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: int Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, ref int Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32

Returns

Int32

Int16(Char, Int32, Int32, Int32, ref Single)

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: float Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single

Returns

Int32

Int16(Char, Int32, Int32, Int32, ref UInt16)

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: ushort Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, ref ushort Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt16

Returns

Int32

Int16(Char, Int32, Int32, Int32, ref UInt32)

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: uint Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, ref uint Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32

Returns

Int32

Int16(Char, Int32, Int32, Int32, Single[])

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: float[] Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

Int16(Char, Int32, Int32, Int32, UInt16[])

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: ushort[] Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, ushort[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt16[]

Returns

Int32

Int16(Char, Int32, Int32, Int32, UInt32[])

Int16 (16 Bit signed)-Zugriff: Datentyp der Variable im PC: uint[] Beschreibung: [Int16\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Int16(char DataArea, int DBNr, int Start, int Cnt, uint[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32[]

Returns

Int32

Int32(Char, Int32, Int32, Int32, Double[])

Int32 (32-Bit signed)-Zugriff: Datentyp der Variable im PC: double[] Beschreibung: [Int32\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int Int32(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

Int32(Char, Int32, Int32, Int32, Int32[])

Int32 (32-Bit signed)-Zugriff: Datentyp der Variable im PC: int[] Beschreibung: Int32(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int Int32(char DataArea, int DBNr, int Start, int Cnt, int[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32[]

Returns

Int32

Int32(Char, Int32, Int32, Int32, ref Double)

Int32 (32-Bit signed)-Zugriff: Datentyp der Variable im PC: double Beschreibung: [Int32\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int Int32(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

[DataArea](#) Char

[DBNr](#) Int32

[Start](#) Int32

[Cnt](#) Int32

[Buf](#) Double

Returns

[Int32](#)

Int32(Char, Int32, Int32, Int32, ref Int32)

Int32 (32-Bit signed)-Zugriff: Datentyp der Variable im PC: int

C#

```
public int Int32(char DataArea, int DBNr, int Start, int Cnt, ref int Buf)
```

Parameters

[DataArea](#) Char

Datenbereich, siehe: [DA](#)

[DBNr](#) Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Int32

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Int32(Char, Int32, Int32, Int32, ref Single)

Int32 (32-Bit signed)-Zugriff: Datentyp der Variable im PC: float Beschreibung: Int32(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int Int32(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single

Returns

Int32

Int32(Char, Int32, Int32, Int32, ref UInt32)

Int32 (32-Bit signed)-Zugriff: Datentyp der Variable im PC: uint Beschreibung: [Int32\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int Int32(char DataArea, int DBNr, int Start, int Cnt, ref uint Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32

Returns

Int32

Int32(Char, Int32, Int32, Int32, Single[])

Int32 (32-Bit signed)-Zugriff: Datentyp der Variable im PC: float[] Beschreibung: [Int32\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int Int32(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

Int32(Char, Int32, Int32, Int32, UInt32[])

Int32 (32-Bit signed)-Zugriff: Datentyp der Variable im PC: uint[] Beschreibung: Int32(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int Int32(char DataArea, int DBNr, int Start, int Cnt, uint[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32[]

Returns

Int32

Int64(Char, Int32, Int32, Int32, Double[])

Int64 (64-Bit signed)-Zugriff: Datentyp der Variable im PC: double[]

C#

```
public int Int64(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

Int64(Char, Int32, Int32, Int32, Int64[])

LINT (64-Bit signed)-Zugriff: Datentyp der Variable im PC: int[]

C#

```
public int Int64(char DataArea, int DBNr, int Start, int Cnt, long[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int64[]

Returns

Int32

Int64(Char, Int32, Int32, Int32, ref Double)

Int64 (64-Bit signed)-Zugriff: Datentyp der Variable im PC: double

C#

```
public int Int64(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double

Returns

Int32

Int64(Char, Int32, Int32, Int32, ref Int64)

LINT (64-Bit signed)-Zugriff: Datentyp der Variable im PC: int

C#

```
public int Int64(char DataArea, int DBNr, int Start, int Cnt, ref long Buf)
```

Parameters

DataArea Char

Datenbereich, siehe: DA

DBNr Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Int64

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Int64(Char, Int32, Int32, Int32, ref Single)

Int64 (64-Bit signed)-Zugriff: Datentyp der Variable im PC: float

C#

```
public int Int64(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single

Returns

Int32

Int64(Char, Int32, Int32, Int32, Single[])

Int64 (64-Bit signed)-Zugriff: Datentyp der Variable im PC: float[]

C#

```
public int Int64(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

LReal(Char, Int32, Int32, Int32, Double[])

LReal (Fließpunktzahl 64-Bit)-Zugriff: Datentyp der Variable im PC: double[]

C#

```
public int LReal(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

LReal(Char, Int32, Int32, Int32, ref Double)

Real (Fließpunktzahl 64-Bit)-Zugriff: Datentyp der Variable im PC: double

C#

```
public int LReal(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double

Returns

Int32

Real(Char, Int32, Int32, Int32, Double[])

Real (Fließpunktzahl 32-Bit)-Zugriff: Datentyp der Variable im PC: double[] Beschreibung: DWord(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int Real(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

Real(Char, Int32, Int32, Int32, ref Double)

Real (Fließpunktzahl 32-Bit)-Zugriff: Datentyp der Variable im PC: double Beschreibung: DWord(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int Real(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double

Returns

Int32

Real(Char, Int32, Int32, Int32, ref Single)

Real (Fließpunktzahl 32-Bit)-Zugriff: Datentyp der Variable im PC: float

C#

```
public int Real(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters

DataArea Char

Datenbereich, siehe: [DA](#)

DBNr Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Single

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Real(Char, Int32, Int32, Int32, Single[])

Real (Fließpunktzahl 32-Bit)-Zugriff: Datentyp der Variable im PC: float[] Beschreibung: [DWord\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int Real(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

SetData(Char, Int32, Int32, Int32, Int32, Int32, PCDataType)

C#

```
public void SetData(char DataArea, int DBNr, int Cnt, int Start, int StartBit, int S7DataType, PCDataType PCDataType)
```

Parameters

DataArea Char

DBNr Int32

Cnt Int32

Start Int32

StartBit Int32

S7DataType Int32

PCDataType PCDataType

String(Char, Int32, Int32, Int32, ref String)

C#

```
public int String(char DataArea, int DBNr, int Start, int Cnt, ref string Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf String

Returns

Int32

Timer(Char, Int32, Int32, Int32, Double[])

Timer Datentyp der Variable im PC: double [] Beschreibung: [Timer\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int Timer(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

Timer(Char, Int32, Int32, Int32, Int32[])

Timer Datentyp der Variable im PC: int[] Beschreibung: Timer(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int Timer(char DataArea, int DBNr, int Start, int Cnt, int[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32[]

Returns

Int32

Timer(Char, Int32, Int32, Int32, ref Double)

Timer Datentyp der Variable im PC: double Beschreibung: Timer(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int Timer(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double

Returns

Int32

Timer(Char, Int32, Int32, Int32, ref Int32)

Timer Datentyp der Variable im PC: int

C#

```
public int Timer(char DataArea, int DBNr, int Start, int Cnt, ref int Buf)
```

Parameters

DataArea Char

Datenbereich, siehe: DA

DBNr Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Int32

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Timer(Char, Int32, Int32, Int32, ref Single)

Timer Datentyp der Variable im PC: float Beschreibung: Timer(Char, Int32, Int32, Int32, Int32%40)

C#

```
public int Timer(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single

Returns

Int32

Timer(Char, Int32, Int32, Int32, ref UInt32)

Timer Datentyp der Variable im PC: uint Beschreibung: `Timer(Char, Int32, Int32, Int32, Int32%40)`

C#

```
public int Timer(char DataArea, int DBNr, int Start, int Cnt, ref uint Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32

Returns

Int32

Timer(Char, Int32, Int32, Int32, Single[])

Timer Datentyp der Variable im PC: float[] Beschreibung: [Timer\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int Timer(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

Timer(Char, Int32, Int32, Int32, UInt32[])

Timer Datentyp der Variable im PC: uint[] Beschreibung: [Timer\(Char, Int32, Int32, Int32, Int32%40\)](#)

C#

```
public int Timer(char DataArea, int DBNr, int Start, int Cnt, uint[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32[]

Returns

Int32

ToString()

C#

```
public override string ToString()
```

Returns

String

UInt64(Char, Int32, Int32, Int32, ref UInt64)

ULINT (64-Bit signed)-Zugriff: Datentyp der Variable im PC: uint

C#

```
public int UInt64(char DataArea, int DBNr, int Start, int Cnt, ref ulong Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt64

Returns

Int32

UInt64(Char, Int32, Int32, Int32, UInt64[])

ulong (64-Bit signed)-Zugriff: Datentyp der Variable im PC: ulong[]

C#

```
public int UInt64(char DataArea, int DBNr, int Start, int Cnt, ulong[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt64[]

Returns

Int32

Word(Char, Int32, Int32, Int32, Char[])

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: char[] Beschreibung: [Word\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, char[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Char[]

Returns

Int32

Word(Char, Int32, Int32, Int32, Double[])

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: double[] Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, double[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double[]

Returns

Int32

Word(Char, Int32, Int32, Int32, Int16[])

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: short[] Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, short[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int16[]

Returns

Int32

Word(Char, Int32, Int32, Int32, Int32[])

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: int[] Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, int[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Int32[]

Returns

Int32

Word(Char, Int32, Int32, Int32, ref Char)

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: char Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, ref char Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Char

Returns

Int32

Word(Char, Int32, Int32, Int32, ref Double)

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: double Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, ref double Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Double

Returns

Int32

Word(Char, Int32, Int32, Int32, ref Int16)

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: short

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, ref short Buf)
```

Parameters

DataArea Char

Datenbereich, siehe: DA

DBNr Int32

Nummer des gewünschten Datenbausteins. Wird nur bei Zugriff auf DB ausgewertet.

Start Int32

Nummer des Bytes, ab welchem der Lesevorgang gestartet werden soll.

Cnt Int32

Anzahl der zu lesenden Elemente.

Buf Int16

Pointer auf den Speicher der Daten im Anwenderprogramm.

Returns

Int32

Auftrag wurde eingetragen, IPS7.Result.E_PC_BUFSIZE (-20) Bei Array, das übergebene Array ist zu klein.

Word(Char, Int32, Int32, Int32, ref Int32)

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: int Beschreibung: [Word\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, ref int Buf)
```

Parameters

[DataArea](#) Char

[DBNr](#) Int32

[Start](#) Int32

[Cnt](#) Int32

[Buf](#) Int32

Returns

[Int32](#)

Word(Char, Int32, Int32, Int32, ref Single)

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: float Beschreibung: [Word\(Char, Int32, Int32, Int32, Int16%40\)](#)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, ref float Buf)
```

Parameters

[DataArea](#) Char

[DBNr](#) Int32

Start Int32

Cnt Int32

Buf Single

Returns

Int32

Word(Char, Int32, Int32, Int32, ref UInt16)

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: ushort Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, ref ushort Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt16

Returns

Int32

Word(Char, Int32, Int32, Int32, ref UInt32)

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: uint Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, ref uint Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32

Returns

Int32

Word(Char, Int32, Int32, Int32, Single[])

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: float[] Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, float[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf Single[]

Returns

Int32

Word(Char, Int32, Int32, Int32, UInt16[])

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: ushort[] Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, ushort[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt16[]

Returns

Int32

Word(Char, Int32, Int32, Int32, UInt32[])

Word (16 Bit)-Zugriff: Datentyp der Variable im PC: uint[] Beschreibung: Word(Char, Int32, Int32, Int32, Int16%40)

C#

```
public int Word(char DataArea, int DBNr, int Start, int Cnt, uint[] Buf)
```

Parameters

DataArea Char

DBNr Int32

Start Int32

Cnt Int32

Buf UInt32[]

Returns

Int32

Table of Contents

Constructors	1
IPS7RdMulti()	1
Properties	1
Cnt	1
DataArea	1
DataType	1
DBNr	2
ErrMemSize	2
PcDataType	2
Result	2
Start	3
StartBit	3
UserData0	3
UserData1	3
Methods	4
Bit(Char, Int32, Int32, Int32, Int32, Byte[])	4
Bit(Char, Int32, Int32, Int32, Int32, Char[])	5
Bit(Char, Int32, Int32, Int32, Int32, Double[])	5
Bit(Char, Int32, Int32, Int32, Int32, Int16[])	6
Bit(Char, Int32, Int32, Int32, Int32, Int32[])	7
Bit(Char, Int32, Int32, Int32, Int32, ref Byte)	7
Bit(Char, Int32, Int32, Int32, Int32, ref Char)	8
Bit(Char, Int32, Int32, Int32, Int32, ref Double)	9
Bit(Char, Int32, Int32, Int32, Int32, ref Int16)	9
Bit(Char, Int32, Int32, Int32, Int32, ref Int32)	10
Bit(Char, Int32, Int32, Int32, Int32, ref SByte)	11
Bit(Char, Int32, Int32, Int32, Int32, ref Single)	11
Bit(Char, Int32, Int32, Int32, Int32, ref UInt16)	12
Bit(Char, Int32, Int32, Int32, Int32, ref UInt32)	13
Bit(Char, Int32, Int32, Int32, Int32, SByte[])	13
Bit(Char, Int32, Int32, Int32, Int32, Single[])	14
Bit(Char, Int32, Int32, Int32, Int32, UInt16[])	15
Bit(Char, Int32, Int32, Int32, Int32, UInt32[])	15
Byte(Char, Int32, Int32, Int32, Byte[])	16
Byte(Char, Int32, Int32, Int32, Char[])	17
Byte(Char, Int32, Int32, Int32, Double[])	17
Byte(Char, Int32, Int32, Int32, Int16[])	18
Byte(Char, Int32, Int32, Int32, Int32[])	18
Byte(Char, Int32, Int32, Int32, ref Byte)	19
Byte(Char, Int32, Int32, Int32, ref Char)	20
Byte(Char, Int32, Int32, Int32, ref Double)	20
Byte(Char, Int32, Int32, Int32, ref Int16)	21
Byte(Char, Int32, Int32, Int32, ref Int32)	21
Byte(Char, Int32, Int32, Int32, ref SByte)	22
Byte(Char, Int32, Int32, Int32, ref Single)	23
Byte(Char, Int32, Int32, Int32, ref UInt16)	23
Byte(Char, Int32, Int32, Int32, ref UInt32)	24
Byte(Char, Int32, Int32, Int32, SByte[])	24
Byte(Char, Int32, Int32, Int32, Single[])	25
Byte(Char, Int32, Int32, Int32, UInt16[])	26
Byte(Char, Int32, Int32, Int32, UInt32[])	26

Counter(Char, Int32, Int32, Int32, Char[])	27
Counter(Char, Int32, Int32, Int32, Double[])	27
Counter(Char, Int32, Int32, Int32, Int16[])	28
Counter(Char, Int32, Int32, Int32, Int32[])	29
Counter(Char, Int32, Int32, Int32, ref Char)	29
Counter(Char, Int32, Int32, Int32, ref Double)	30
Counter(Char, Int32, Int32, Int32, ref Int16)	30
Counter(Char, Int32, Int32, Int32, ref Int32)	31
Counter(Char, Int32, Int32, Int32, ref Single)	32
Counter(Char, Int32, Int32, Int32, ref UInt16)	32
Counter(Char, Int32, Int32, Int32, ref UInt32)	33
Counter(Char, Int32, Int32, Int32, Single[])	34
Counter(Char, Int32, Int32, Int32, UInt16[])	34
Counter(Char, Int32, Int32, Int32, UInt32[])	35
DWord(Char, Int32, Int32, Int32, Double[])	35
DWord(Char, Int32, Int32, Int32, Int32[])	36
DWord(Char, Int32, Int32, Int32, ref Double)	37
DWord(Char, Int32, Int32, Int32, ref Int32)	37
DWord(Char, Int32, Int32, Int32, ref Single)	38
DWord(Char, Int32, Int32, Int32, ref UInt32)	39
DWord(Char, Int32, Int32, Int32, Single[])	39
DWord(Char, Int32, Int32, Int32, UInt32[])	40
GetData(Byte[])	40
GetData(Char[])	41
GetData(Double[])	41
GetData(Int16[])	42
GetData(Int32[])	42
GetData(Int64[])	43
GetData(ref Byte)	43
GetData(ref Char)	43
GetData(ref Double)	44
GetData(ref Int16)	44
GetData(ref Int32)	45
GetData(ref Int64)	45
GetData(ref SByte)	45
GetData(ref Single)	46
GetData(ref String)	46
GetData(ref UInt16)	47
GetData(ref UInt32)	47
GetData(ref UInt64)	47
GetData(SByte[])	48
GetData(Single[])	48
GetData(UInt16[])	49
GetData(UInt32[])	49
GetData(UInt64[])	49
Int16(Char, Int32, Int32, Int32, Char[])	50
Int16(Char, Int32, Int32, Int32, Double[])	50
Int16(Char, Int32, Int32, Int32, Int16[])	51
Int16(Char, Int32, Int32, Int32, Int32[])	52
Int16(Char, Int32, Int32, Int32, ref Char)	52
Int16(Char, Int32, Int32, Int32, ref Double)	53
Int16(Char, Int32, Int32, Int32, ref Int16)	53
Int16(Char, Int32, Int32, Int32, ref Int32)	54

Int16(Char, Int32, Int32, Int32, ref Single)	55
Int16(Char, Int32, Int32, Int32, ref UInt16)	55
Int16(Char, Int32, Int32, Int32, ref UInt32)	56
Int16(Char, Int32, Int32, Int32, Single[])	57
Int16(Char, Int32, Int32, Int32, UInt16[])	57
Int16(Char, Int32, Int32, Int32, UInt32[])	58
Int32(Char, Int32, Int32, Int32, Double[])	58
Int32(Char, Int32, Int32, Int32, Int32[])	59
Int32(Char, Int32, Int32, Int32, ref Double)	60
Int32(Char, Int32, Int32, Int32, ref Int32)	60
Int32(Char, Int32, Int32, Int32, ref Single)	61
Int32(Char, Int32, Int32, Int32, ref UInt32)	62
Int32(Char, Int32, Int32, Int32, Single[])	62
Int32(Char, Int32, Int32, Int32, UInt32[])	63
Int64(Char, Int32, Int32, Int32, Double[])	63
Int64(Char, Int32, Int32, Int32, Int64[])	64
Int64(Char, Int32, Int32, Int32, ref Double)	65
Int64(Char, Int32, Int32, Int32, ref Int64)	65
Int64(Char, Int32, Int32, Int32, ref Single)	66
Int64(Char, Int32, Int32, Int32, Single[])	66
LReal(Char, Int32, Int32, Int32, Double[])	67
LReal(Char, Int32, Int32, Int32, ref Double)	68
Real(Char, Int32, Int32, Int32, Double[])	68
Real(Char, Int32, Int32, Int32, ref Double)	69
Real(Char, Int32, Int32, Int32, ref Single)	69
Real(Char, Int32, Int32, Int32, Single[])	70
SetData(Char, Int32, Int32, Int32, Int32, Int32, PCDataType)	71
String(Char, Int32, Int32, Int32, ref String)	71
Timer(Char, Int32, Int32, Int32, Double[])	72
Timer(Char, Int32, Int32, Int32, Int32[])	73
Timer(Char, Int32, Int32, Int32, ref Double)	73
Timer(Char, Int32, Int32, Int32, ref Int32)	74
Timer(Char, Int32, Int32, Int32, ref Single)	74
Timer(Char, Int32, Int32, Int32, ref UInt32)	75
Timer(Char, Int32, Int32, Int32, Single[])	76
Timer(Char, Int32, Int32, Int32, UInt32[])	76
ToString()	77
UInt64(Char, Int32, Int32, Int32, ref UInt64)	77
UInt64(Char, Int32, Int32, Int32, UInt64[])	78
Word(Char, Int32, Int32, Int32, Char[])	78
Word(Char, Int32, Int32, Int32, Double[])	79
Word(Char, Int32, Int32, Int32, Int16[])	79
Word(Char, Int32, Int32, Int32, Int32[])	80
Word(Char, Int32, Int32, Int32, ref Char)	81
Word(Char, Int32, Int32, Int32, ref Double)	81
Word(Char, Int32, Int32, Int32, ref Int16)	82
Word(Char, Int32, Int32, Int32, ref Int32)	83
Word(Char, Int32, Int32, Int32, ref Single)	83
Word(Char, Int32, Int32, Int32, ref UInt16)	84
Word(Char, Int32, Int32, Int32, ref UInt32)	84
Word(Char, Int32, Int32, Int32, Single[])	85
Word(Char, Int32, Int32, Int32, UInt16[])	86
Word(Char, Int32, Int32, Int32, UInt32[])	86