

Abstract

This document describes a CSharp interface to [SWI-Prolog](#). The described interface provides a layer around the C-interface for natural programming from C#. The interface deals with automatic type-conversion to and from Prolog, mapping of exceptions and making queries to Prolog in an easy way. There is a call-back from Prolog to C#.

Introduction

The first version of this Interface was more or less a port of the C++ interface. Now the naming is more '.Net' like and the interface provides number of features that make queries to SWI-Prolog very easy and powerful. Using programmable type-conversion (casting), native data-types can be translated automatically into appropriate Prolog types. Automatic destruction deals with most of the cleanup required.

Acknowledgements

I would like to thank Jan Wielemaker for answering many questions and for his comments.

Also to Arne Skjærholt for the 64-Bit version (SwiPICs64.dll) and Batu Akan for the Mono code. Foutelet Joel provide the F# sample.

Download binaries

Here is the link to download the latest binaries or older versions.

[Download page](#)

At present I only publish the binaries including the documentation on this site.

The sources, which are under LGPL 2, are on [GitHub/SWI-Prolog/contrib-swiplcs](#).

Versions

The latest version work with SWI-Prolog 6.3.1 and higher.

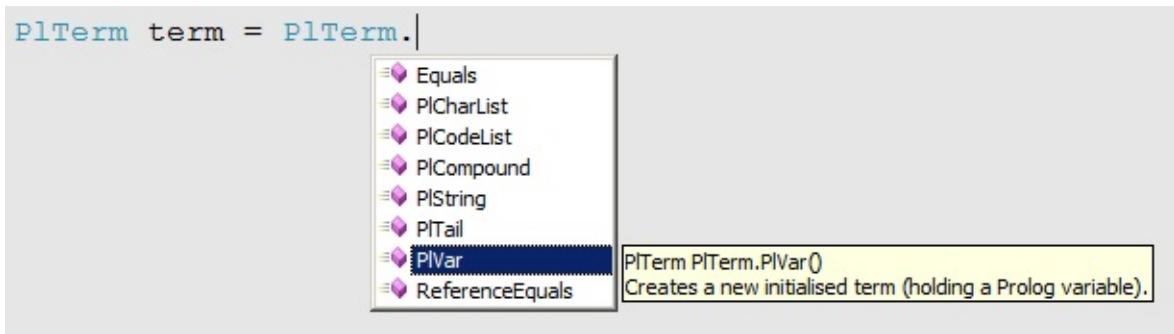
The AssemblyVersion number, e.g. 1.1.60601.0, can be interpreted as follows:

- 1 - major version
- 1 - minor version
- 60601 - SWI-Prolog version 6.6.1 (test cases run against this prolog version)
- 0 - patch level version

Getting started

Copy SwiPICs.dll or SwiPICs64.dll and SwiPICs.XML where ever you want and add a project reference to SwiPICs.dll.

After that IntelliSense and tool tips should be available.



Make sure that libswipl.dll and its dependencies could be found by the system. For the sample below it is: libswipl.dll, pthreadGC2.dll, libgmp-10.dll, files.dll For a big application it could be a lot more.

TIP: For development add the SWI-prolog bin directory to the PATH environment variable.

NOTE: Don't forget to restart Visual Studio after that. VS must recognize the new environment for debugging.

Basically windows search first in the folder where the executable resides than in the windows system directory and at least in the directories that

are listed in the PATH environment variable. For details see "["Dynamic-Link Library Search Order"](#)"

If libswipl.dll or one of its dependencies could not found you will receive a error like

System.IO.FileNotFoundException: Das angegebene Modul wurde nicht gefunden. (Ausnahme von HRESULT: 0x8007007E)

An other common error is:

SWI-Prolog: [FATAL ERROR:
 Could not find system resources]
Failed to release stacks

To fix this add the SWI_HOME_DIR environment variable as described [SWI-Prolog FAQ FindResources](#) with a statement like this before calling PlEngine.Initialize.

```
Environment.SetEnvironmentVariable("SWI_HOME_DIR",  
    @"the_PATH_to_boot32.prc");
```

First program

A sample says more then I want to write here.

C#

 Cop

```
using System;  
using SbsSW.SwiPlCs;  
  
namespace HelloWorldDemo  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            //Environment.SetEnvironmentVariable("Sw  
            if (!PlEngine.IsInitialized)  
            {  
                String[] param = { "-q" }; // suppr
```

```
    PlEngine.Initialize(param);
    PlQuery.PlCall("assert(father(martin, g));
    PlQuery.PlCall("assert(father(uwe, g));
    PlQuery.PlCall("assert(father(uwe, m));
    PlQuery.PlCall("assert(father(uwe, a));
    using (var q = new PlQuery("father(P, L)"))
    {
        foreach (PlQueryVariables v in q)
            Console.WriteLine(v["L"].ToString());
    }
    Console.WriteLine("all children of uwe");
    q.Variables["P"].Unify("uwe");
    foreach (PlQueryVariables v in q)
        Console.WriteLine(v["C"].ToString());
    }
    PlEngine.PlCleanup();
    Console.WriteLine("finished!");
}
}
```

Here is how to use the library in F#. Manny thanks to Foutelet Joel for this sample.

F#

Cop

```
// Learn more about F# at http://fsharp.net
open System
open SbsSW.SwiPlCs;

let ple = PlEngine.IsInitialized in
    if ple then printfn "Echeck initialisation" else
        begin
            PlEngine.Initialize(["-q"])
            PlQuery.PlCall("assert(father(martin, ian))")
            PlQuery.PlCall("assert(father(uwe, gloria))")
            PlQuery.PlCall("assert(father(uwe, mela))")
            PlQuery.PlCall("assert(father(uwe, ayal))")
```

```

let q = new PlQuery "father(P, C), atom
begin
    Seq.iter (fun (x : PlQueryVariables
        printfn "all children from uwe:"
        let r : PlQueryVariables = q.Variables
        Seq.iter (fun (x : PlQueryVariables
    end

    PlEngine.PlCleanup()
    printfn "%A" "finished!"
end

```

For further samples see the examples in [SbsSW.SwiPICs](#) and [SbsSW.SwiPICs.PIEngine](#).

The class [SbsSW.SwiPICs.PIQuery](#) is the key to ask SWI-Prolog for proofs or solutions.

The [SbsSW.SwiPICs.PITerm](#) plays a central role in conversion and operating on Prolog data.

Programming tips

I strongly recomanate to use [SbsSW.SwiPICs.PIQuery](#) in a **using statement** using statement like in the sample program above.
An alternative is to call [Dispose](#) explicitly like in the sample below.

C#

 Cop

```

const string strRef = "a;e;";
PlQuery.PlCall("assert(n('" + strRef + "'))");
var q = new PlQuery("n(X)");
Assert.IsTrue(q.NextSolution());
Assert.AreEqual(strRef, q.Variables["X"].ToString());
var q2 = new PlQuery("n('" + strRef + "')");
Assert.IsTrue(q2.NextSolution());
Assert.AreEqual(strRef, q.Variables["X"].ToString())

```

```
q2.Dispose();  
q.Dispose();
```

Note that access to the query Variables is impossible after Dispose().

Known Bugs

[SbsSW.SwiPICs.PIEngine.Initialize](#) work *not* as expected if there are e.g. German umlauts in the parameters e.g. in the path or filename for a qlf file (switch -x)

See marshalling in the source NativeMethods.cs

by Uwe Lesta, SBS-Softwaresysteme GmbH



The online documentation home is [here](#).

This namespace SbsSW.SwiPICs provides an .NET interface to [SWI-Prolog](#)

Overview

Prolog variables are dynamically typed and all information is passed around using the C-interface type term_t which is an int. In C#, term_t is embedded in the lightweight struct [PITerm](#). Constructors and operator definitions provide flexible operations and integration with important C#-types ([string](#), [int](#) and [double](#)).

The list below summarises the important classes / struct defined in the C# interface.

class / struct	Short description
PIEngine	A static class represents the prolog engine.
PITerm	A struct representing prolog data.
PITermV	A vector of PITerm .
PIQuery	A class to query Prolog.

Types

All Types	Classes	Structures	Enumerations
Icon	Type	Description	
	PIEngine	This static class represents the prolog engine.	



[PIFrame](#)

The class PIFrame provides an interface to discard unused term-references as well as rewinding unifications (data-backtracking). Reclaiming unused term-references is automatically performed after a call to a C#-defined predicate has finished and returns control to Prolog. In this scenario PIFrame is rarely of any use.

This class comes into play if the top level program is defined in C# and calls Prolog multiple times. Setting up arguments to a query requires term-references and using PIFrame is the only way to reclaim them.



[PIQuery](#)

This class allows queries to prolog.

A query can be created by a string or by constructing compound terms see [Constructors](#) for details.

All resources and terms created by a query are reclaimed by [Dispose\(\)](#). It is recommended to build a query in a using scope.

There are four possible opportunities to query Prolog

Query type	Description
A static call	To ask prolog for

		a proof. Return only true or false.
	A PICallQuery(String)	To get the first result of a goal
	Construct a PIQuery object by a string.	The most convenient way.
	Construct a PIQuery object by compound terms.	The most flexible and fast (runtime) way.
	For examples see PIQuery(String) and PIQuery(String, PITermV)	
	PIQuerySwitch	Flags that control for the foreign predicate parameters SWI-Prolog Manual - 9.6.16 Querying Prolog .
	PIQueryVar	Represents one variable of a Query result.
	PIQueryVariables	Represents the set variables of a Query if it was created from a string. This class is also used to represent the results of a PIQuery after ToList() or SolutionVariables was called.



[PITerm](#)

The PIterm struct plays a central role in conversion and operating on Prolog data.

PITerm implements [IComparable](#) to support ordering in [\[!:\]System.Linq](#) queries if PIterm is a List.

Creating a PIterm can be done by the [Constructors](#) or by the following static methods:

`PIVar()`, `PITail()`, `PICompound`,
`PIString()`, `PICodeList()`, `PICharList()`
(see remarks)



[PITermV](#)

The struct PItermv represents an array of term-references.

This type is used to pass the argument to a foreign defined predicate (see [DelegateParameterVarArgs](#)), construct compound terms (see [PICompound\(String, PItermV\)](#)) and to create queries (see [PIQuery](#)).

The only useful member function is the overloading of `[]`, providing (0-based) access to the elements. [Item\[Int32\]](#)
Range checking is performed and raises a [ArgumentOutOfRangeException](#) exception.



PIType

Obtain the type of a term, which should be a term returned by one of the other interface predicates or passed as an argument. The function returns the type of the Prolog term. The type identifiers are listed below.

Examples

Before going into a detailed description of the CSharp classes let me present a few examples illustrating the 'feel' of the interface. The Assert class in the sample is from the test framework and has nothing to do with the interface. It shows only which return values are expected.

Creating terms

This very simple example shows the basic creation of a Prolog term and how a Prolog term is converted to C#-data:

[Copy](#)

```
PlTerm t1 = new PlTerm("x(A)");
PlTerm t2 = new PlTerm("x(1)");
Assert.IsTrue(t1.Unify(t2));
Assert.AreEqual("x(1)", t1.ToString());
```

Calling Prolog

This example shows how to make a simple call to prolog.

[Copy](#)

```
PlTerm l1 = new PlTerm("[a,b,c,d]");
Assert.IsTrue(PlQuery.PlCall("is_list", l1));
```

Getting the solutions of a query

This example shows how to obtain all solutions of a prolog query.

[PlQuery](#) takes the name of a predicate and the goal-argument vector as arguments. From this information it deduces the arity and locates the predicate. the member-function NextSolution() yields true if there was a solution and false otherwise. If the goal yielded a Prolog exception it is mapped into a C# exception.

```
PlQuery q = new PlQuery("member", new PlTermV  
while (q.NextSolution())  
    Console.WriteLine(s[0].ToString());
```

There is an other constructor of [PlQuery](#) which simplify the sample above.

[Cop](#)

```
PlQuery q = new PlQuery("member(A, [a,b,c])")  
foreach (PlTermV s in q.Solutions)  
    Console.WriteLine(s[0].ToString());
```

An other way to get the results is to use [SolutionVariables](#) to iterate over [PlQueryVariables](#).

[Cop](#)

```
PlQuery q = new PlQuery("member(A, [a,b,c])")  
foreach (PlQueryVariables vars in q.SolutionV  
    Console.WriteLine(vars["A"].ToString());
```

It is also possible to get all solutions in a list by [ToList\(\)](#). This could be used to work with LinQ to objects which is really nice. [PlQuery](#) and

[ToListAsync](#) for further samples.

```
var results = from n in new PIQuery("member(A  
foreach (var s in results)  
    Console.WriteLine(s.A);
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

PIEngine

Class

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#)

C# ▾

This static class represents the prolog engine.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static class P1Engine  
  
Public NotInheritable Class P1Engine  
  
public ref class P1Engine abstract sealed  
  
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type P1Engine = class end
```

Members

All Members	Methods	Properties	
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited

Icon	Member	Description
	Initialize(String[])	Initialise SWI-Prolog

 IsInitialized	To test if the prolog engine has been initialized.
 PICleanup()	Try a clean up but it is better to use the web interface for "possible regression". From PI 5.4.7 to pl-5.6.27" to see what has changed.
 PIHalt()	Stops the PIEngine and releases all resources.
 PIThreadAttachEngine()	<p>return : reference count</p> <p>If an error occurs, -1 is returned.</p> <p>If this Prolog is not compatible with the current thread, -2 is returned.</p>
 PIThreadDestroyEngine()	<p>Destroy the Prolog engine attached to the thread. Only takes effect if PL_thread_attach() is called as many times as PL_thread_detach() in this thread.</p> <p>Please note that construction and destruction of engines are expensive operations. Consider using a single engine if performance is critical or memory is a critical resource.</p>
 PIThreadSelf()	This method is also provided for the single-threaded version.

		where it returns -2.
= 	RegisterForeign(Delegate)	Register a C# callback
= 	RegisterForeign(String, Delegate)	Register a C# callback
= 	RegisterForeign(String, Int32, Delegate)	Register a C# callback
= 	RegisterForeign(String, String, Int32, Delegate)	Register a C# callback
= 	SetStreamFunctionRead(PIStreamType, DelegateStreamReadFunction)	TODO
= 	SetStreamFunctionWrite(PIStreamType, DelegateStreamWriteFunction)	This is a primitive approach to output from a stream.

- Examples

A sample

[Copy](#)

```
if (!P1Engine.IsInitialized)
{
    String[] empty_param = { "" };
    P1Engine.Initialize(empty_param);
```

```

    // do some funny things ...
    PlEngine.PlCleanup();
}
// program ends here

```

The following sample show how a file is consult via comand-line options

```

public void Demo_consult_pl_file_by_param()
{
    string[] ref_values = { "gloria", "melanie" };
    Console.WriteLine("Demo_consult_pl_file_by_param");

    // Build a prolog source file (skip this part)
    string filename = Path.GetTempFileName();
    StreamWriter sw = File.CreateText(filename);
    sw.WriteLine("father(martin, inka).");
    sw.WriteLine("father(uwe, gloria).");
    sw.WriteLine("father(uwe, melanie).");
    sw.WriteLine("father(uwe, ayala).");
    sw.Close();

    // build the parameterstring to Initialize
    String[] param = { "-q", "-f", filename };
    try
    {
        PlEngine.Initialize(param);
        Console.WriteLine("all child's from uwe are ");
        using (PlQuery q = new PlQuery("father(uwe, ?)"))
        {
            int idx = 0;
            foreach (PlQueryVariables v in q)
            {
                Console.WriteLine(v);
            }
        }
    }
}
```

```
        Console.WriteLine(v["Child"]);
        Assert.AreEqual(ref_values[id], v["Child"]);
    }
}
catch (PlException e)
{
    Console.WriteLine(e.MessagePl);
    Console.WriteLine(e.Message);
}
finally
{
    PlEngine.PlCleanup();
}
} // Demo_consult_pl_file_by_param
```

- Inheritance Hierarchy

[Object](#)

└ [PIEngine](#)

- See Also

[SbsSW.SwiPICs.Callback](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method (argv)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

Initialize(String[])

C# ▾

Initialise SWI-Prolog

The write method of the output stream is redirected to [SbsSW.SwiPICs.Streams](#) before Initialize. The read method of the input stream just after Initialize.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static void Initialize(
    string[] argv
)

Public Shared Sub Initialize (
    argv As String()
)

public:
static void Initialize(
    array<String^>^ argv
)

static member Initialize :
    argv : string[] -> unit
```

Parameters

argv (String[])

For a complete parameter description see the [SWI-Prolog reference](#)

[manual section 2.4 Command-line options.](#)

sample parameter:

```
String[] param = { "-q", "-f", @"some\file
```

At the first position a parameter "" is added in this method.

[PL_initialise](#)

■ Remarks

A known bug: Initialize work *not* as expected if there are e.g. German umlauts in the parameters See marshalling in the source NativeMethods.cs

■ Examples

For an example see [PIEngine](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **IsInitialized Property**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ► **IsInitialized**

C# ▾

To test if the prolog engine is up.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool IsInitialized { get; }

Public Shared ReadOnly Property IsInitialized
    Get

public:
static property bool IsInitialized {
    bool get ();
}

static member IsInitialized : bool with get
```

Property Value

[Boolean](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PlCleanup Method**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlEngine](#) ► [PlCleanup\(\)](#)

C# ▾

Try a clean up but it is buggy search the web for "possible regression from pl-5.4.7 to pl-5.6.27" to see reasons

▀ Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static void PlCleanup()  
  
Public Shared Sub PlCleanup  
  
public:  
static void Plcleanup()  
  
static member Plcleanup : unit -> unit
```

▀ Remarks

Use this method only at the last call before run program ends

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ► **PlHalt()**

C# ▾

Stops the PIEngine and **the program**

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static void PlHalt()  
  
Public Shared Sub PlHalt  
  
public:  
static void PlHalt()  
  
static member PlHalt : unit -> unit
```

Remarks

SWI-Prolog calls internally pl_cleanup and than exit(0)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PIThreadAttachEngine Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ► [PIThreadAttachEngine\(\)](#)

C# ▾

return : reference count of the engine

If an error occurs, -1 is returned.

If this Prolog is not compiled for multi-threading, -2 is returned.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static int PIThreadAttachEngine()  
  
Public Shared Function PIThreadAttachEngine A  
  
public:  
    static int PIThreadAttachEngine()  
  
    static member PIThreadAttachEngine : unit ->
```

Return Value

[Int32](#)

A reference count of the engine

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



PIThreadDestroyEngine Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

PIThreadDestroyEngine()

C# ▾

Destroy the Prolog engine in the calling thread. Only takes effect if `PL_thread_destroy_engine()` is called as many times as `PL_thread_attach_engine()` in this thread.

Please note that construction and destruction of engines are relatively expensive operations. Only destroy an engine if performance is not critical and memory is a critical resource.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool PIThreadDestroyEngine()  
  
Public Shared Function PIThreadDestroyEngine  
public:  
    static bool PIThreadDestroyEngine()  
  
    static member PIThreadDestroyEngine : unit ->
```

Return Value

[Boolean](#)

Returns `true` on success and `false` if the calling thread has no engine or this Prolog does not support threads.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



PIThreadSelf Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ► [PIThreadSelf\(\)](#)

C#

This method is also provided in the single-threaded version of SWI-Prolog, where it returns -2.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static int PIThreadSelf()  
  
Public Shared Function PIThreadSelf As Integer  
  
public:  
    static int PIThreadSelf()  
  
    static member PIThreadSelf : unit -> int
```

Return Value

[Int32](#)

Returns the integer Prolog identifier of the engine or -1 if the calling thread has no Prolog engine.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **RegisterForeign** Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

RegisterForeign()

C# 

Register a C#-function to implement a Prolog predicate.

After this call returns successfully, a predicate with name (a string) and arity arity (a C# int) is created in module module.

If module is NULL, the predicate is created in the module of the calling context or if no context is present in the module user.

■ Remarks

Add a additional namespace by:

using SbsSW.SwiPICs.Callback;

■ Examples

For an example see [DelegateParameter2](#) and [DelegateParameter1](#).

■ Members

Icon	Member	Description
	RegisterForeign(Delegate)	Register a C# callback method
	RegisterForeign(String, Delegate)	Register a C# callback method
	RegisterForeign(String, Int32, Delegate)	Register a C# callback method

	<u>RegisterForeign(String, String, Int32, Delegate)</u>	Register a C# callback method
---	---	-------------------------------

See Also

[SbsSW.SwiPICs.Callback](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

RegisterForeign Method (method)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

[RegisterForeign\(Delegate\)](#)

C# ▾

Register a C# callback method

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool RegisterForeign(  
    Delegate method  
)  
  
Public Shared Function RegisterForeign (  
    method As Delegate  
) As Boolean  
  
public:  
static bool RegisterForeign(  
    Delegate^ method  
)  
  
static member RegisterForeign :  
    method : Delegate -> bool
```

Parameters

method ([Delegate](#))

a delegate to a c# method [SbsSW.SwiPICs.Callback](#)

Return Value

[Boolean](#)

true if registration succeed otherwise false

Examples

For an example see [DelegateParameter2](#) and [DelegateParameter1](#).

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **RegisterForeign Method (module, method)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

RegisterForeign(String, Delegate)

C# ▾

Register a C# callback method

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool RegisterForeign(
    string module,
    Delegate method
)
```

```
Public Shared Function RegisterForeign (
    module As String,
    method As Delegate
) As Boolean
```

```
public:
static bool RegisterForeign(
    String^ module,
    Delegate^ method
)
```

```
static member RegisterForeign :
    module : string *
    method : Delegate -> bool
```

Parameters

***module* ([String](#))**

the name of a prolog module [Using Modules](#)

method ([Delegate](#))

a delegate to a c# method [SbsSW.SwiPICs.Callback](#)

■ Return Value

[Boolean](#)

true if registration succeed otherwise false

■ Examples

For an example see [DelegateParameter2](#) and [DelegateParameter1](#).

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **RegisterForeign Method (name, arity, method)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

RegisterForeign(String, Int32, Delegate)

C# ▾

Register a C# callback method

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool RegisterForeign(
    string name,
    int arity,
    Delegate method
)
```

```
Public Shared Function RegisterForeign (
    name As String,
    arity As Integer,
    method As Delegate
) As Boolean
```

```
public:
static bool RegisterForeign(
    String^ name,
    int arity,
    Delegate^ method
)
```

```
static member RegisterForeign :
    name : string *
    arity : int *
```

method : Delegate -> bool

[-] Parameters

***name* ([String](#))**

The name of a static C# method

***arity* ([Int32](#))**

The amount of parameters

***method* ([Delegate](#))**

a delegate to a c# method [SbsSW.SwiPICs.Callback](#)

[-] Return Value

[Boolean](#)

true if registration succeed otherwise false

[-] Examples

For an example see [DelegateParameterVarArgs](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **RegisterForeign Method (module, name, arity, method)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

RegisterForeign(String, String, Int32, Delegate)

C# ▾

Register a C# callback method

[-] Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool RegisterForeign(
    string module,
    string name,
    int arity,
    Delegate method
)
```

```
Public Shared Function RegisterForeign (
    module As String,
    name As String,
    arity As Integer,
    method As Delegate
) As Boolean
```

```
public:
static bool RegisterForeign(
    String^ module,
    String^ name,
    int arity,
    Delegate^ method
)
```

```
static member RegisterForeign :  
    module : string *  
    name : string *  
    arity : int *  
    method : Delegate -> bool
```

[-] Parameters

module ([String](#))

The name of the module (Prolog module system)

name ([String](#))

The name of a static C# method

arity ([Int32](#))

The amount of parameters

method ([Delegate](#))

a delegate to a c# method [SbsSW.SwiPICs.Callback](#)

[-] Return Value

[Boolean](#)

true if registration succeed otherwise false

[-] Examples

For an example see [DelegateParameterVarArgs](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

SetStreamFunctionRead Method

(streamType, function)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

**SetStreamFunctionRead(PIStreamType,
DelegateStreamReadFunction)**

C# ▾

TODO

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static void SetStreamFunctionRead(  
    PIStreamType streamType,  
    DelegateStreamReadFunction function  
)  
  
Public Shared Sub SetStreamFunctionRead (  
    streamType As PIStreamType,  
    function As DelegateStreamReadFunction  
)  
  
public:  
static void SetStreamFunctionRead(  
    PIStreamType streamType,  
    DelegateStreamReadFunction^ function  
)  
  
static member SetStreamFunctionRead :  
    streamType : PIStreamType *  
    function : DelegateStreamReadFunction
```

Parameters

streamType ([PIStreamType](#))

Determine which stream to use [PIStreamType](#)

function (DelegateStreamReadFunction)
A [DelegateStreamReadFunction](#)

[-] Examples

```
private const string ValidationStringRead = ""  
  
static internal long Sread(IntPtr handle, Int  
{  
    const string s = ValidationStringRead + ""  
    byte[] array = System.Text.Encoding.Unico  
    System.Runtime.InteropServices.Marshal.Co  
    return array.Length;  
}  
  
  
[TestMethod]  
public void StreamRead()  
{  
    var rf = new DelegateStreamReadFunction(S  
    PlEngine.SetStreamFunctionRead(PIStreamTy  
    // NOTE: read/1 needs a dot ('.') at the  
    PlQuery.PlCall("assert( (test_read(A) :-  
    PlTerm t = PlQuery.PlCallQuery("test_read  
    Assert.AreEqual(ValidationStringRead, t.T  
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0

(1.1.60605.0)



SetStreamFunctionWrite Method

(streamType, function)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIEngine](#) ►

**SetStreamFunctionWrite(PIStreamType,
DelegateStreamWriteFunction)**

C# ▾

This is a primitive approach to enter the output from a stream.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static void SetStreamFunctionWrite(  
    PlStreamType streamType,  
    DelegateStreamWriteFunction function  
)  
  
Public Shared Sub SetStreamFunctionWrite (  
    streamType As PlStreamType,  
    function As DelegateStreamWriteFunction  
)  
  
public:  
static void SetStreamFunctionWrite(  
    PlStreamType streamType,  
    DelegateStreamWriteFunction^ function  
)  
  
static member SetStreamFunctionWrite :  
    streamType : PlStreamType *  
    function : DelegateStreamWriteFunction
```

Parameters

streamType ([PlStreamType](#))

Determine which stream to use [PlStreamType](#)

function (DelegateStreamWriterFunction)
A [DelegateStreamWriterFunction](#)

[- Examples

```
static string _testString;

static long Swrite(IntPtr handle, string buff
{
    string s = buffer.Substring(0, (int)buffe
    _testString = s;
    return buffersize;
}

[TestMethod]
public void Streamwrite()
{
    // NOTE: the Swrite function is only call
    const string validationString = "Hello .n
    PlQuery.PlCall("assert( (test_write :- wr
    var wf = new DelegateStreamWriterFunction(
        PlEngine.SetStreamFunctionWrite(PlStreamT
        PlQuery.PlCall("test_write");
        Assert.AreEqual(validationString + "\r\n",
    }
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Class[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► **PIFrame**

C# ▾

The class PIFrame provides an interface to discard unused term-references as well as rewinding unifications (data-backtracking). Reclaiming unused term-references is automatically performed after a call to a C#-defined predicate has finished and returns control to Prolog. In this scenario PIFrame is rarely of any use.

This class comes into play if the top level program is defined in C# and calls Prolog multiple times. Setting up arguments to a query requires term-references and using PIFrame is the only way to reclaim them.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public class PIFrame : IDisposable

Public Class PIFrame
    Implements IDisposable

public ref class PIFrame : IDisposable

type PIFrame =
    class
        interface IDisposable
    end
```

Members

All Members

Constructors

Methods

 Public
 Protected

 Instance
 Static

 Declared

Icon	Member	Description	Inherited From
	PIFrame()	Creating an instance of this class marks all term-references created afterwards to be valid only in the scope of this instance.	
	Dispose()	Implements IDisposable.	
	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)	
	Finalize()	Reclaims all term-references created after constructing the instance. (Overrides Object.Finalize() .)	
	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object .)	
	GetType()	Gets the type of the current instance. (Inherited from Object .)	
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)	
	Rewind()	Discards all term-references and global-stack data created as well as undoing all unifications after the instance was created.	

	ToString()	<p>Returns a string that represents the current object. (Inherited from Object.)</p>
---	----------------------------	---

[-] Remarks

see <http://www.swi-prolog.org/pldoc/package/pl2cpp.html#sec:8.1>

[-] Examples

A typical use for PlFrame is the definition of C# methods that call Prolog and may be called repeatedly from C#. Consider the definition of assertWord(), adding a fact to word/1:

```
void AssertWord2(string word)
{
    PlFrame fr = new PlFrame();
    PlTermV av = new PlTermV(1);
    av[0] = PlTerm.PlCompound("word", new PlT
    PlQuery q = new PlQuery("assert", av);
    q.NextSolution();
    q.Dispose(); // IMPORTANT ! never forge
    fr.Dispose();
}
```

alternatively you can use

```
void AssertWord(string word)
{
    using (PlFrame fr = new PlFrame())
    {
        PlTermV av = new PlTermV(1);
```

```
    av[0] = PlTerm.PlCompound("word", new
    using (PlQuery q = new PlQuery("asser
    {
        q.NextSolution();
    }
}
```



Caution: NOTE: in any case you have to destroy any query object used inside a PIFrame



by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIFrame](#) ► [PIFrame\(\)](#)

C# ▾

Creating an instance of this class marks all term-references created afterwards to be valid only in the scope of this instance.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIFrame()

Public Sub New

public:
PIFrame()

new : unit -> PIFrame
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Dispose Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIFrame](#) ► [Dispose\(\)](#)

C# ▾

Implement IDisposable.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public void Dispose()

Public Sub Dispose

public:
virtual void Dispose() sealed

abstract Dispose : unit -> unit
override Dispose : unit -> unit
```

Remarks

Do not make this method virtual.

A derived class should not be able to override this method.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIFrame](#) ► **Finalize()**

C# ▾

Reclaims all term-references created after constructing the instance.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
protected override void Finalize()
```

```
Protected Overrides Sub Finalize
```

```
protected:
```

```
virtual void Finalize() override
```

```
abstract Finalize : unit -> unit
```

```
override Finalize : unit -> unit
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIFrame](#) ► **Rewind()**

C# ▾

Discards all term-references and global-stack data created as well as undoing all unifications after the instance was created.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public void Rewind()

Public Sub Rewind

public:
void Rewind()

member Rewind : unit -> unit
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Class

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► **PIQuery**

C# ▾

This class allows queries to prolog.

A query can be created by a string or by constructing compound terms see [Constructors](#) for details.

All resources an terms created by a query are reclaimed by [Dispose\(\)](#). It is recommended to build a query in a using scope.

There are four possible opportunities to query Prolog

Query type	Description
A static call	To ask prolog for a proof. Return only true or false.
A PICallQuery(String)	To get the first result of a goal
Construct a PIQuery object by a string.	The most convenient way.
Construct a PIQuery object by compound terms.	The most flexible and fast (runtime) way.

For examples see [PIQuery\(String\)](#) and [PIQuery\(String, PITermV\)](#)

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public class PIQuery : IDisposable
Public Class PIQuery
    Implements IDisposable
```

```

public ref class PIQuery : IDisposable
type PIQuery =
    class
        interface IDisposable
    end

```

Members

All Members	Constructors	Methods	Properties
<input checked="" type="checkbox"/> Public <input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited	
Icon	Member	Description	
	PIQuery(String)	<p>With this constructor a query is created from a string.</p> <p>Uppercase parameters are interpreted as variables but can't be nested in subterms. If you need a variable in a nested term use PIQuery(String, PITermV). See the examples for details.</p>	
	PIQuery(String, String)	locating the predicate in the named module.	
	PIQuery(String, PITermV)	Create a query where name defines the name of the predicate and av the argument vector. The arity is deduced from av. The predicate is located in	

		the Prolog module user.
≡	PIQuery(String, String, PITermV)	locating the predicate in the named module.
Args	Args	Provide access to the Argument vector for the query
≡	Dispose()	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
💡	Dispose(Boolean)	Release all resources from the query
≡	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
💡	Finalize()	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Overrides Object.Finalize() .)
≡	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object .)
≡	GetType()	Gets the type of the current instance.

		(Inherited from Object .)
💡	MemberwiseClone()	Creates a shallow copy of the current Object .
		(Inherited from Object .)
✳️	NextSolution()	Provide the next solution to the query Prolog exceptions are mapped to C# exceptions.
✳️ S	PICall(String, PITermV)	Create a query where name defines the name of the predicate and av the argument vector. The arity is deduced from av. The predicate is located in the Prolog module user.
✳️ S	PICall(String, String, PITermV)	As PICall(String, PITermV) but locating the predicate in the named module.
✳️ S	PICall(String)	Call a goal once.
✳️ S	PICallQuery(String)	<p>NOTE:will be changed in the near future.</p> <p>return the solution of a query which is called once by call Throw an ArgumentException if there is no or more than one variable in the goal</p>

 PICallQuery(String, String)	As PICallQuery(String) but executed in the named module.
 Query(PIQuerySwitch)	<p>Obtain status information on the Prolog system. The actual argument type depends on the information required. The parameter queryType describes what information is wanted</p> <p>Returning pointers and integers as a long is bad style. The signature of this function should be changed.</p> <p>PIQuerySwitch</p>
 Solutions	<p>Enumerate the solutions.</p> <p>For examples see PIQuery(String)</p>
 SolutionVariables	Enumerate the PIQueryVariables of one solution.
 ToList()	<p>Create a ReadOnlyCollection<T> of PIQueryVariables.</p> <p>If calling <code>ToList()</code> all solutions of the query are generated and stored in the Collection.</p>

 ToString()	Returns a string that represents the current object. (Inherited from Object .)
 VariableNames	Gets a Collection<T> of the variable names if the query was built by a string.
 Variables	The List of PIQueryVariables of this PIQuery.

[-] Remarks

The query will be opened by [NextSolution\(\)](#) and will be closed if NextSolution() return false.

[-] Inheritance Hierarchy

[Object](#)

└ **PIQuery**

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ► [PIQuery\(\)](#)

C# ▾

With these constructors a Prolog query can be created but not opened.
To get the results see [NextSolution\(\)](#)

A Query can be created from a string or by a name and PITermV. The later is a native way and available for compatibility.

If a Query is created from a string representing arbitrary prolog text the helper classes [PIQueryVar](#) and [PIQueryVariables](#) comes into the game. In this case the most convenient way to get the results is to use [SolutionVariables](#) or [ToList\(\)](#).

For examples see [PIQuery\(String\)](#).

- Members

Icon	Member	Description
	PIQuery(String)	With this constructor a query is created from a string. Uppercase parameters are interpreted as variables but can't be nested in subterms. If you need a variable in a nested term use PIQuery(String, PItermV) . See the examples for details.
	PIQuery(String, String)	locating the predicate in the named module.

	<u>PIQuery(String, PITermV)</u>	Create a query where name defines the name of the predicate and av the argument vector. The arity is deduced from av. The predicate is located in the Prolog module user.
	<u>PIQuery(String, String, PITermV)</u>	locating the predicate in the named module.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor (goal)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► [PlQuery\(String\)](#)

C# ▾

With this constructor a query is created from a string.

Uppercase parameters are interpreted as variables but can't be nested in sub terms. If you need a variable in a nested term use [PlQuery\(String, PlTermV\)](#). See the examples for details.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlQuery(
    string goal
)

Public Sub New (
    goal As String
)

public:
PlQuery(
    String^ goal
)

new :
    goal : string -> PlQuery
```

Parameters

goal (String)

A string for a prolog query

Remarks

Muddy Waters sang:"I'am build for comfort, I ain't build for speed"

[- Examples

```
public void QueryStringForeach()
{
    string[] mm = { "aa", "bb", "cc" };
    var q = new PlQuery("member(A, [aa, bb, c
    int i = 0;
    foreach (PlTermV s in q.Solutions)
    {
        Assert.AreEqual(mm[i++], s[0].ToString
    }
    // or with named variables
    i = 0;
    foreach (PlQueryVariables s in q.Solution
    {
        Assert.AreEqual(mm[i++], s["A"].ToString
    }
}
```

This sample shows a query with two variables.

```
public void QueryString2()
{
    var q = new PlQuery("append(A, B, [a,b,c]
    Assert.IsTrue(q.NextSolution());
    Assert.AreEqual("[ ]", q.Args[0].ToString(
    Assert.AreEqual("[a,b,c]", q.Args[1].ToSt
    Assert.IsTrue(q.NextSolution());
    Assert.AreEqual("[a]", q.Args[0].ToString
```

```
        Assert.AreEqual("[b,c]", q.Args[1].ToString()
}
```

And the same with named variables.

```
public void QueryStringNamed()
{
    var q = new PlQuery("append(A, B, [a,b,c]")
    Assert.IsTrue(q.NextSolution());
    Assert.AreEqual("[]", q.Variables["A"].ToString());
    Assert.AreEqual("[a,b,c]", q.Variables["B"].ToString());
    Assert.IsTrue(q.NextSolution());
    Assert.AreEqual("[a]", q.Variables["A"].ToString());
    Assert.AreEqual("[b,c]", q.Variables["B"].ToString()
}
```

This sample shows what happens if the argument vector is used with compound terms.

```
public void PlCallQueryCompound_string()
{
    string[] mm = { "comp(aa,aa1)", "comp(aa,bb1)" };
    build_pred(); // create: test(comp(X,Y))
    var q = new PlQuery("test(comp(aa,X)))");
    int i = 0;
    foreach (PlTermV s in q.Solutions)
    {
        Assert.AreEqual(mm[i++], s[0].ToString()
    }
}
```

```
}
```

And here how to get the results with named variables with compound terms.

```
public void PlCallQueryCompoundNamed_string()
{
    string[] mm = { "aa1", "aa2", "aa3" };
    build_pred(); // create: test(comp(X,Y))
    var q = new PlQuery("test(comp(aa,X))");
    int i = 0;
    foreach (PlQueryVariables v in q.Solution)
    {
        Assert.AreEqual(mm[i++], v["X"].ToString())
    }
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog **PlQuery**

Constructor (name, termV)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► [PlQuery\(String, PlTermV\)](#)

C# ▾

Create a query where name defines the name of the predicate and av the argument vector. The arity is deduced from av. The predicate is located in the Prolog module user.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlQuery(  
    string name,  
    PlTermV termV  
)  
  
Public Sub New (  
    name As String,  
    termV As PlTermV  
)  
  
public:  
PlQuery(  
    String^ name,  
    PlTermV termV  
)  
  
new :  
    name : string *  
    termV : PlTermV -> PlQuery
```

Parameters

***name* (String)**

the name of the predicate

***termV* ([PlTermV](#))**

the argument vector containing the parameters

[-] Examples

This sample shows a query with a compound term as an argument.

```
public void PlCallQueryCompound_termv()
{
    string[] mm = { "aa1", "aa2", "aa3" };
    build_pred(); // create: test(comp(X,Y))
    PlTerm var1 = PlTerm.PlVar();
    PlTerm comp = PlTerm.PlCompound("comp", n
        using (var q = new PlQuery("test", new Pl
    {
        int i = 0;
        foreach (PlTermV s in q.Solutions)
        {
            Assert.AreEqual(mm[i++], var1.ToString());
            Assert.AreEqual(comp.ToString(),
        }
    }
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog **PlQuery**

Constructor (module, goal)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► [PlQuery\(String, String\)](#)

C# ▾

locating the predicate in the named module.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlQuery(  
    string module,  
    string goal  
)  
  
Public Sub New (  
    module As String,  
    goal As String  
)  
  
public:  
PlQuery(  
    String^ module,  
    String^ goal  
)  
  
new :  
    module : string *  
    goal : string -> PlQuery
```

Parameters

***module* (String)**

locating the predicate in the named module.

goal (String)

A string for a prolog query

Remarks

Muddy Waters sang:"I'am build for comfort, I ain't build for speed"

Examples

```
public void QueryStringForeach()
{
    string[] mm = { "aa", "bb", "cc" };
    var q = new PlQuery("member(A, [aa, bb, c
    int i = 0;
    foreach (PlTermV s in q.Solutions)
    {
        Assert.AreEqual(mm[i++], s[0].ToString()
    }
    // or with named variables
    i = 0;
    foreach (PlQueryVariables s in q.Solution
    {
        Assert.AreEqual(mm[i++], s["A"].ToString()
    }
}
```

This sample shows a query with two variables.

```
public void QueryString2()
{
    var q = new PlQuery("append(A, B, [a,b,c]
    Assert.IsTrue(q.NextSolution());
    Assert.AreEqual("[", q.Args[0].ToString()
```

```

        Assert.AreEqual("[a,b,c]", q.Args[1].ToString());
        Assert.IsTrue(q.NextSolution());
        Assert.AreEqual("[a]", q.Args[0].ToString());
        Assert.AreEqual("[b,c]", q.Args[1].ToString());
    }

```

And the same with named variables.

```

public void QueryStringNamed()
{
    var q = new PlQuery("append(A, B, [a,b,c])");
    Assert.IsTrue(q.NextSolution());
    Assert.AreEqual("[]", q.Variables["A"].ToString());
    Assert.AreEqual("[a,b,c]", q.Variables["B"].ToString());
    Assert.IsTrue(q.NextSolution());
    Assert.AreEqual("[a]", q.Variables["A"].ToString());
    Assert.AreEqual("[b,c]", q.Variables["B"].ToString());
}

```

This sample shows what happens if the argument vector is used with compound terms.

```

public void PlCallQueryCompound_string()
{
    string[] mm = { "comp(aa,aa1)", "comp(aa,bb1)" };
    build_pred(); // create: test(comp(X,Y))
    var q = new PlQuery("test(comp(aa,X))");
    int i = 0;
    foreach (PlTermV s in q.Solutions)
    {

```

```
        Assert.AreEqual(mm[i++], s[0].ToString()
    }
}
```

And here how to get the results with named variables with compound terms.

```
public void PlCallQueryCompoundNamed_string()
{
    string[] mm = { "aa1", "aa2", "aa3" };
    build_pred(); // create: test(comp(X,Y)
    var q = new PlQuery("test(comp(aa,X))");
    int i = 0;
    foreach (PlQueryVariables v in q.Solution
    {
        Assert.AreEqual(mm[i++], v["X"].ToString()
    }
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta%20at%20SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog **PlQuery**

Constructor (module, name, termV)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► [PlQuery\(String, String, PlTermV\)](#)

C# ▾

locating the predicate in the named module.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlQuery(  
    string module,  
    string name,  
    PlTermV termV  
)  
  
Public Sub New (  
    module As String,  
    name As String,  
    termV As PlTermV  
)  
  
public:  
PlQuery(  
    String^ module,  
    String^ name,  
    PlTermV termV  
)  
  
new :  
    module : string *  
    name : string *  
    termV : PlTermV -> PlQuery
```

[-] Parameters

module ([String](#))

locating the predicate in the named module.

name ([String](#))

the name of the predicate

termV ([PlTermV](#))

the argument vector containing the parameters

[-] Examples

This sample shows a query with a compound term as an argument.

```
public void PlCallQueryCompound_termv()
{
    string[] mm = { "aa1", "aa2", "aa3" };
    build_pred(); // create: test(comp(X,Y)
    PlTerm var1 = PlTerm.PlVar();
    PlTerm comp = PlTerm.PlCompound("comp", n
        using (var q = new PlQuery("test", new Pl
    {
        int i = 0;
        foreach (PlTermV s in q.Solutions)
        {
            Assert.AreEqual(mm[i++], var1.ToS
            Assert.AreEqual(comp.ToString(),
        }
    }
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► **Args**

C# ▾

Provide access to the Argument vector for the query

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTermV Args { get; }

Public ReadOnly Property Args As PlTermV
    Get

public:
property PlTermV Args {
    PlTermV get ();
}

member Args : PlTermV with get
```

Property Value

[PlTermV](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Dispose Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ► **Dispose()**

C# ▾

▀ Members

Icon	Member	Description
	Dispose()	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
	Dispose(Boolean)	Release all resources from the query

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ► **Dispose()**

C# ▾

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public void Dispose()

Public Sub Dispose

public:
virtual void Dispose() sealed

abstract Dispose : unit -> unit
override Dispose : unit -> unit
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (disposing)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ► **Dispose(Boolean)**

C# ▾

Release all resources from the query

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
protected virtual void Dispose(
    bool disposing
)

Protected Overridable Sub Dispose (
    disposing As Boolean
)

protected:
virtual void Dispose(
    bool disposing
)

abstract Dispose :
    disposing : bool -> unit
override Dispose :
    disposing : bool -> unit
```

Parameters

***disposing* (Boolean)**

if true all is deleted

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ► **Finalize()**

C# ▾

Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
protected override void Finalize()
```

```
Protected Overrides Sub Finalize
```

```
protected:
```

```
virtual void Finalize() override
```

```
abstract Finalize : unit -> unit
```

```
override Finalize : unit -> unit
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **NextSolution** Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ► [NextSolution\(\)](#)

C#

Provide the next solution to the query. Prolog exceptions are mapped to C# exceptions.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool NextSolution()

Public Function NextSolution As Boolean

public:
bool NextSolution()

member NextSolution : unit -> bool
```

Return Value

[Boolean](#)

return true if successful and false if there are no (more) solutions.

Remarks

If the query is closed it will be opened. If the last solution was generated the query will be closed.

If an exception is thrown while parsing (open) the query the _qid is set to zero.

Exceptions

Exception	Condition
PIException	Is thrown if SWI-Prolog Manual PL_next_solution() returns false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► [PICall\(\)](#)

C# ▾

The main purpose of the static PlCall methods is to call a prolog procedure to do some site effects.

▀ Examples

```
Assert.IsTrue(PlQuery.PICall("is_list", new PIList("a", "b", "c")));
```

```
Assert.IsTrue(PlQuery.PICall("consult", new PIString("user")));  
// or
```

```
Assert.IsTrue(PlQuery.PICall("consult('some_file')"));
```

▀ Members

Icon	Member	Description
	PICall(String, PITermV)	Create a query where name defines the name of the predicate and av the argument vector. The arity is deduced from av. The predicate is located in the Prolog module user.
	PICall(String, String, PITermV)	As PICall(String, PITermV) but locating the predicate in the named module.
	PICall(String)	Call a goal once.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method (goal)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► **PlCall(String)**

C# ▾

Call a goal once.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool PlCall(
    string goal
)

Public Shared Function PlCall (
    goal As String
) As Boolean

public:
static bool PlCall(
    String^ goal
)

static member PlCall :
    goal : string -> bool
```

Parameters

goal ([String](#))

The complete goal as a string

Return Value

[Boolean](#)

Return true or false as the result of NextSolution() or throw an exception

Remarks

Create a PIQuery from the arguments, generates the first solution by NextSolution() and destroys the query.

■ Examples

```
Assert.IsTrue(PIQuery.PICall("is_list([a,b,c,
```

```
Assert.IsTrue(PIQuery.PICall("consult('some_f
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method (predicate, args)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► **PlCall(String, PlTermV)**

C# ▾

Create a query where name defines the name of the predicate and av the argument vector. The arity is deduced from av. The predicate is located in the Prolog module user.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool PlCall(
    string predicate,
    PlTermV args
)

Public Shared Function PlCall (
    predicate As String,
    args As PlTermV
) As Boolean

public:
static bool PlCall(
    String^ predicate,
    PlTermV args
)

static member PlCall :
    predicate : string *
    args : PlTermV -> bool
```

Parameters

***predicate* (String)**

defines the name of the predicate

args ([PITermV](#))

Is a [PITermV](#) of arguments for the predicate

[-] Return Value

[Boolean](#)

Return true or false as the result of NextSolution() or throw an exception

[-] Remarks

Create a PIQuery from the arguments, generates the first solution by NextSolution() and destroys the query.

[-] Examples

This sample shows a query with a compound term as an argument.

```
public void PlCallQueryCompound_termv()
{
    string[] mm = { "aa1", "aa2", "aa3" };
    build_pred(); // create: test(comp(X,Y)
    PlTerm var1 = PlTerm.PlVar();
    PlTerm comp = PlTerm.PlCompound("comp", n
    using (var q = new PlQuery("test", new Pl
    {
        int i = 0;
        foreach (PlTermV s in q.Solutions)
        {
            Assert.AreEqual(mm[i++], var1.ToS
            Assert.AreEqual(comp.ToString(),
        }
    }
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog **PlCall**

Method (module, predicate, args)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► **PlCall(String, String, PlTermV)**

C# ▾

As [PlCall\(String, PlTermV\)](#) but locating the predicate in the named module.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool PlCall(
    string module,
    string predicate,
    PlTermV args
)

Public Shared Function PlCall (
    module As String,
    predicate As String,
    args As PlTermV
) As Boolean

public:
static bool PlCall(
    String^ module,
    String^ predicate,
    PlTermV args
)

static member PlCall :
    module : string *
    predicate : string *
    args : PlTermV -> bool
```

[-] Parameters

module ([String](#))

locating the predicate in the named module.

predicate ([String](#))

defines the name of the predicate

args ([PlTermV](#))

Is a [PlTermV](#) of arguments for the predicate

[-] Return Value

[Boolean](#)

Return true or false as the result of NextSolution() or throw an exception

[-] Remarks

Create a PlQuery from the arguments, generates the first solution by NextSolution() and destroys the query.

[-] Examples

This sample shows a query with a compound term as an argument.

```
public void PlCallQueryCompound_termv()
{
    string[] mm = { "aa1", "aa2", "aa3" };
    build_pred(); // create: test(comp(X,Y)
    PlTerm var1 = PlTerm.PlVar();
    PlTerm comp = PlTerm.PlCompound("comp", n
    using (var q = new PlQuery("test", new Pl
    {
        int i = 0;
        foreach (PlTermV s in q.Solutions)
        {
```

```
        Assert.AreEqual(mm[i++], var1.ToString());
        Assert.AreEqual(comp.ToString(),
    }
}
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PICallQuery Method**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ► **PICallQuery()**

C# ▾

▀ Members

Icon	Member	Description
 S	PICallQuery(String)	<p>NOTE:will be changed in the near future.</p> <p>return the solution of a query which is called once by call Throw an ArgumentException if there is no or more than one variable in the goal</p>
 S	PICallQuery(String, String)	As PICallQuery(String) but executed in the named module.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PICallQuery Method (goal)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ►

PICallQuery(String)

C# ▾

NOTE:will be changed in the near future.

return the solution of a query which is called once by call Throw an ArgumentException if there is no or more than one variable in the goal

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PIterm PlCallQuery(  
        string goal  
)  
  
Public Shared Function PlCallQuery (  
        goal As String  
) As PIterm  
  
public:  
static PIterm PlCallQuery(  
        String^ goal  
)  
  
static member PlCallQuery :  
        goal : string -> PIterm
```

Parameters

***goal* (String)**

a goal with *one* variable

Return Value

PITerm

the bound variable of the first solution

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PlCallQuery Method (module, goal)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ►

PlCallQuery(String, String)

C# ▾

As [PlCallQuery\(String\)](#) but executed in the named module.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PlTerm PlCallQuery(
    string module,
    string goal
)

Public Shared Function PlCallQuery (
    module As String,
    goal As String
) As PlTerm

public:
static PlTerm PlCallQuery(
    String^ module,
    String^ goal
)

static member PlCallQuery :
    module : string *
    goal : string -> PlTerm
```

Parameters

***module* ([String](#))**

The modulename in which the query is executed

***goal* ([String](#))**

a goal with **one** variable

▀ Return Value

PITerm

the bound variable of the first solution

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (queryType)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ►

Query(PlQuerySwitch)

C# ▾

Obtain status information on the Prolog system. The actual argument type depends on the information required. The parameter queryType describes what information is wanted.

Returning pointers and integers as a long is bad style. The signature of this function should be changed.

PlQuerySwitch

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static long Query(
    PlQuerySwitch queryType
)

Public Shared Function Query (
    queryType As PlQuerySwitch
) As Long

public:
static long long Query(
    PlQuerySwitch queryType
)

static member Query :
    queryType : PlQuerySwitch -> int64
```

Parameters

***queryType* ([PlQuerySwitch](#))**

A PlQuerySwitch.

▀ Return Value

[Int64](#)

A int depending on the given queryType

▀ Examples

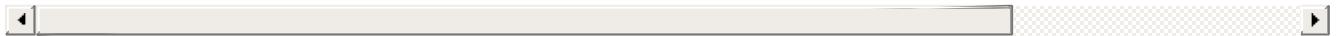
This sample shows how to get SWI-Prologs version number

```
public void Pl_query_version()
{
    long v = PlQuery.Query(PlQuerySwitch.Vers
    Assert.AreEqual(60605, v, "SWI-Prolog ver
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► **Solutions**

C# ▾

Enumerate the solutions.

For examples see [PlQuery\(String\)](#)

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public IEnumerable<PlTermV> Solutions { get;
    Public ReadOnly Property Solutions As IEnumerable<PlTermV>
        Get
    End Property
}

public:
property IEnumerable<PlTermV>^ Solutions {
    IEnumerable<PlTermV>^ get () {
    }
}

member Solutions : IEnumerable<PlTermV> with
```

Property Value

[IEnumerable<PlTermV>](#)

See Also

[NextSolution\(\)](#)

[Constructors](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0

(1.1.60605.0)



SolutionVariables Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► [SolutionVariables](#)

C# ▾

Enumerate the [PlQueryVariables](#) of one solution.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public IEnumerable<PlQueryVariables> SolutionVariables { get; }
```

```
Public ReadOnly Property SolutionVariables As IEnumerable<PlQueryVariables>
```

```
public:
property IEnumerable<PlQueryVariables^>^ SolutionVariables { get; }
```

```
member SolutionVariables : IEnumerable<PlQueryVariables> with get
```

Property Value

[IEnumerable<PlQueryVariables>](#)

Examples

```
public void TestCompoundQuery()
{
    string[] refValues = { "gloria", "melanie" };
    using (new PlFrame())
    {
        PlQuery.PlCall("assert(father(uwe, gloria))");
    }
}
```

```
PlQuery.PlCall("assert(father(uwe, me
PlQuery.PlCall("assert(father(uwe, ay
var plq = new PlQuery("father(P,C), a
int i = 0;
foreach (PlQueryVariables vars in plq
{
    Assert.AreEqual("uwe", (string)va
    Assert.AreEqual(refValues[i++], (
}
}
}
```

See Also

[NextSolution\(\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► [ToList\(\)](#)

C# ▾

Create a [ReadOnlyCollection<T>](#) of [PlQueryVariables](#).

If calling `ToList()` all solutions of the query are generated and stored in the Collection.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public IReadOnlyCollection<PlQueryVariables> T
Public Function ToList As IReadOnlyCollection(
public:
    IReadOnlyCollection<PlQueryVariables^>^ ToList
member ToList : unit -> IReadOnlyCollection<Pl
```

Return Value

[ReadOnlyCollection<PlQueryVariables>](#)

A `IReadOnlyCollection` of `PlQueryVariables` containing all solutions of the query.

Examples

```
public void Test_multi_goal_ToList()
{
    var results = from n in new PlQuery("L=[a
        select new { A = (string)n[
```

```
int i = 0;
foreach (var t in results)
    Assert.AreEqual(_abc[i++], t.A);
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **VariableNames** Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuery](#) ► [VariableNames](#)

C# ▾

Gets a [Collection<T>](#) of the variable names if the query was built by a string.

[-] Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public Collection<string> VariableNames { get  
Public ReadOnly Property VariableNames As Col  
    Get  
  
public:  
property Collection<String^>^ VariableNames {  
    Collection<String^>^ get ();  
}  
  
member VariableNames : Collection<string> wit
```

[-] Property Value

[Collection<String>](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Variables

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQuery](#) ► [Variables](#)

C# ▾

The List of [PlQueryVariables](#) of this PlQuery.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlQueryVariables Variables { get; }

Public ReadOnly Property Variables As PlQuery
    Get

public:
property PlQueryVariables^ Variables {
    PlQueryVariables^ get ();
}

member Variables : PlQueryVariables with get
```

Property Value

[PlQueryVariables](#)

Examples

In the following example you see how the query Variables can be used to set a variable.

```
public void TestCompoundQueryWithVariables()
{
    string[] refValues = { "x_a", "x_b", "x_c"
        using (new PlFrame())
```

```
        {
            var plq = new PlQuery("member(X, [a,b"));
            plq.Variables["P"].Unify("x_");
            int i = 0;
            foreach (PlQueryVariables vars in plq
            {
                Assert.AreEqual("x_", (string)vars["X"]);
                Assert.AreEqual(refValues[i++], (int)vars["Y"]);
            }
        }
    }
```

Here is a more complex sample where the variables of two queries are connected.

```
public void TestCompoundNestedQueryWithVariab
{
    string[] refValues = { "x_a", "x_b", "x_c"
    string[] refValuesInner = { "a1", "a2", "
        int innerIdx = 0;
        using (new PlFrame())
        {
            var plq = new PlQuery("member(X, [a,b
            plq.Variables["P"].Unify("x_");
            int i = 0;
            foreach (PlQueryVariables vars in plq
            {
                Assert.AreEqual("x_", (string)var
                Assert.AreEqual(refValues[i++], (
                    var q = new PlQuery("member(X, [1
                    q.Variables["P"].Unify(plq.Variables["P"]
                    q.Variables["X"].Value = "a1";
                    var result = q.Execute();
                    Assert.AreEqual("a1", result);
                }
            }
        }
    }
}
```

```
        var results = from n in q.ToList()
                      foreach (var v in results)
{
    Assert.AreEqual(refValuesInne
}
}
}
```

See Also

[PIQueryVariables](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



PIQuerySwitch Enumeration

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQuerySwitch](#)

C# ▾

Flags that control for the foreign predicate parameters

[SWI-Prolog Manual - 9.6.16 Querying Prolog](#).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public enum PIQuerySwitch
Public Enumeration PIQuerySwitch
public enum class PIQuerySwitch
type PIQuerySwitch
```

Members

Member	Description
None	The default value.
Argc	Return an integer holding the number of arguments given to Prolog from Unix.
Argv	Return a char ** holding the argument vector given to Prolog from Unix.
GetChar	Read character from terminal.
MaxInteger	Return a long, representing the maximal integer value represented by a Prolog integer.
MinInteger	Return a long, representing the minimal integer value.
Version	Return a long, representing the version as

	$10,000 \times M + 100 \times m + p$, where M is the major, m the minor version number and p the patch-level. For example, 20717 means 2.7.17
MaxThreads	Return the maximum number of threads that can be created in this version. Return values of PL_thread_self() are between 0 and this number.
Encoding	Return the default stream encoding of Prolog (of type IOENC).
UserCpu	Get amount of user CPU time of the process in milliseconds.

- See Also

[Query\(PIQuerySwitch\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PlQueryVar Class**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► **PlQueryVar**

C# ▾

Represents one variable of a Query result.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public class PlQueryVar
Public Class PlQueryVar
public ref class PlQueryVar
type PlQueryVar = class end
```

Members

All Members

Methods

Properties

Public

Instance

Declared

Protected

Static

Inherited

Icon	Member	Description
≡	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
💡	Finalize()	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
≡	GetHashCode()	Serves as a hash function for a

		particular type. (Inherited from Object .)
	GetType()	Gets the type of the current instance. (Inherited from Object .)
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Name	The name of a variable in a Query
	ToString()	Returns a string that represents the current object. (Inherited from Object .)
	Value	The Value (PITerm) of a variable in a Query

Inheritance Hierarchy

[Object](#)

└ **PIQueryVar**

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta@SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQueryVar](#) ► **Name**

C# ▾

The name of a variable in a Query

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public string Name { get; internal set; }

Public Property Name As String
    Get
    Friend Set

public:
property String^ Name {
    String^ get ();
    internal: void set (String^ value);
}

member Name : string with get, internal set
```

Property Value

[String](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQueryVar](#) ► **Value**

C# ▾

The Value (PITerm) of a variable in a Query

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIterm Value { get; internal set; }

Public Property Value As PIterm
    Get
    Friend Set

public:
property PIterm Value {
    PIterm get ();
    internal: void set (PIterm value);
}

member Value : PIterm with get, internal set
```

Property Value

[PITerm](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PIQueryVariables Class**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQueryVariables](#)

C# ▾

Represents the set variables of a Query if it was created from a string.

This class is also used to represent the results of a PIQuery after [ToList](#) or [SolutionVariables](#) was called.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public class PIQueryVariables
Public Class PIQueryVariables
public ref class PIQueryVariables
type PIQueryVariables = class end
```

Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declare

Inherit

Icon

Member

Description



[PIQueryVariables\(\)](#)

Initializes a new instance of the **PIQueryVariables** class



[Count](#)

Returns the number of elements in the sequence. (Defined by Enumerable `c` `List<PIQueryVar>`.)

	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object .)
	GetType()	Gets the type of the current instance. (Inherited from Object .)
	Item[String]	Gets the PITerm of the given variable name or throw an ArgumentException .
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	ToString()	Returns a string that represents the current object. (Inherited from Object .)

Examples

This sample shows both [Variables](#) is used to unify the variables of two nested queries and the result

```
public void TestCompoundNestedQueryWithVariab
{
    string[] refValuesInner = { "_gap_abc1",
    int innerIdx = 0;
    using (new PlFrame())
    {
        try
        {
            var outerQuery = new PlQuery("app
outerQuery.Variables["C"].Unify(n
var innerQuery = new PlQuery("mem
innerQuery.Variables["X"].Unify(o
innerQuery.Variables["Y"].Unify(")
innerQuery.Variables["Z"].Unify(o
foreach (PlQueryVariables vars in
{
    var results = from n in inner
                    select new { L = n["ATOM"]
foreach (var v in results)
{
    Assert.AreEqual(refValues
}
}
}
catch (Exception ex)
{
    System.Diagnostics.Trace.WriteLine
    Assert.IsFalse(true);
}
}
}
```

[-] Inheritance Hierarchy

[Object](#)

└ [PIQueryVariables](#)

[-] See Also

[Variables](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PlQueryVariables** Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQueryVariables](#) ►

[PlQueryVariables\(\)](#)

C# ▾

Initializes a new instance of the [PlQueryVariables](#) class

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlQueryVariables()  
  
Public Sub New  
  
public:  
PlQueryVariables()  
  
new : unit -> PlQueryVariables
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PIQueryVariables](#) ► **Count**

C#

Returns the number of elements in the sequence. (Defined by Enumerable of List<PIQueryVar>.)

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public int Count { get; }

Public ReadOnly Property Count As Integer
    Get

public:
property int Count {
    int get ();
}

member Count : int with get
```

Property Value

[Int32](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property (name)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlQueryVariables](#) ►

Item[String]

C#

Gets the [PlTerm](#) of the given variable name or throw an ArgumentException.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTerm this[
    string name
] { get; }

Public ReadOnly Default Property Item (
    name As String
) As PlTerm
    Get

public:
property PlTerm default[String^ name] {
    PlTerm get (String^ name);
}

member Item : PlTerm with get
```

Parameters

name (String)

The name of the variable

Return Value

[PlTerm](#)

The PlTerm (value) of the variable

Exceptions

Exception	Condition
ArgumentException	Is thrown if the name is not the name of a variable.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Structure

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► **PITerm**

C# ▾

struct plays a central role in conversion and operating on

ements [IComparable](#) to support ordering in [\[!:System.Linq\]](#)
Term is a List.

PITerm can be done by the [Constructors](#) or by the following
ands:

Nil(), PICompound, PIString(), PICodeList(), PICharList() (see

tion Syntax

Visual Basic	Visual C++	F#
Visual Basic	Visual C++	F#

```
struct PITerm : IComparable, IEnumerable<PITerm>
    IEnumerable
```

```
Structure PITerm
    Implements IComparable, IEnumerable(Of PITerm)
    IEnumerable
```

```
value class PITerm : IComparable,
    IEnumerable<PITerm>, IEnumerable
```

```
dAttribute>]
```

```
Term =
```

```
uct
```

```
    interface IComparable
    interface IEnumerable<PITerm>
    interface IEnumerable
```

S

s	Constructors	Methods	Properties
		<input checked="" type="checkbox"/> Instance <input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Declared <input checked="" type="checkbox"/> Inherited
ber		Description	
m(String)		Creates a term-references holding a Prolog term representing text.	
m(Int32)		Creates a term-references holding a Prolog integer representing value.	
m(Double)		Creates a term-references holding a Prolog float representing value.	
?ITerm)		Appends an element to a list by creating a new one and copy all elements Note This is a slow version see my mail from Jan from 2007.11.06 14:44	
ist(PITerm)		Appends a list (PI tail) to a list by creating a new one and copy all elements	

<code>nd(PITerm)</code>	Appends element to the list and make the PI <code>Tail</code> reference point to the new variable tail. If A is a variable, and this method is called on it using the argument "gnat", a list of the form [gnat B] is created and the PI <code>Tail</code> object now points to the new variable B. This method returns TRUE if the unification succeeded and FALSE otherwise. No exceptions are generated.
<code>:0</code>	Get the arity of the functor if PITerm is a compound term.
<code>oareTo(Object)</code>	Unifies the term with [] and returns the result of the unification.
<code>lity(PITerm, PITerm)</code>	Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.
	Yields TRUE if the PI <code>Term</code> is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an

	<code>type_error</code> exception if the conversion failed.
<code>lity(PITerm, Int32)</code>	Yields TRUE if the PIterm is an atom or string representing the same integer as the argument, FALSE if the conversion was not successful. conversion of the term is done by <code>PL_get_long</code>
<code>lity(Int32, PIterm)</code>	Yields TRUE if the PIterm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an <code>type_error</code> exception if the conversion failed.
<code>lity(PITerm, String)</code>	Yields TRUE if the PIterm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an <code>type_error</code> exception if the conversion failed.
<code>lity(String, PIterm)</code>	Yields TRUE if the PIterm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an

	<p>type_error exception if the conversion failed.</p>
Is(Object)	<p>Indicates whether this instance and a specified object are equal.</p> <p>(Overrides ValueType.Equals(Object).)</p>
Cit(PITerm to String)	<p>Converts the Prolog argument into a string which implies Prolog atoms and strings are converted to the represented text or throw a PITypeException.</p>
Cit(PITerm to Int32)	<p>Yields a int if the PIterm is a Prolog integer or float that can be converted without loss to a int. Throws a PITypeException exception otherwise</p>
Cit(PITerm to Double)	<p>Yields the value as a C# double if PIterm represents a Prolog integer or float. Throws a PITypeException exception otherwise.</p>
Ze()	<p>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</p> <p>(Inherited from Object.)</p>

numerator()	Returns an enumerator that iterates through the collection.
hashCode()	Returns the hash code for this instance. (Overrides ValueType.GetHashCode() .)
GetType()	Gets the type of the current instance. (Inherited from Object .)
Equal(PITerm, PIAtom)	Yields TRUE if the PIAtom is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an <code>type_error</code> exception if the conversion failed.
EqualOrEqual(PITerm, PIAtom)	Yields TRUE if the PIAtom is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an <code>type_error</code> exception if the conversion failed.
Equality(PITerm, PIAtom)	Yields TRUE if the PIAtom is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an <code>type_error</code> exception if the conversion failed.

	<code>type_error</code> exception if the conversion failed.
arity(PITerm, Int32)	summary
arity(Int32, PITerm)	summary
arity(PITerm, String)	test
arity(String, PITerm)	test
atom	Return true if PITerm is an atom.
atomic	Return true if PITerm is atomic (not variable or compound).
compound	Return true if PITerm is a compound term. Note that a list is a compound term ./2
float	Return true if PITerm is a float.
ground	Return true if PITerm is a ground term. See also <code>ground/1</code> . This function is cycle-safe.

alized	return false for a PIterm variable which is only declared and true if it is also initialized
ger	Return true if PITerm is an integer.
	Return true if PITerm is a compound term with functor ./2 or the atom [].
nber	Return true if PITerm is an integer or float.
ng	Return true if PITerm is a string.
	Return true if PITerm is a variable
nt32]	<p>If PIterm is compound and index is between 0 and Arity (including), the nth PIterm is returned.</p> <p>If pos is 0 the functor of the term is returned (For a list '.').</p> <p>See: PL_get_arg/3</p>

Than(PITerm, PITerm)	Yields TRUE if the PITerm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.
ThanOrEqual(PITerm, m)	Yields TRUE if the PITerm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.
otherwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
name()	Get a holding the name of the functor if PITerm is a compound term.
/value()	return a PITerm bound to the next element of the list PITail and advance PITail. Returns the element on success or a free PITerm (Variable) if PITail represents the empty list. If PITail is neither a list nor the empty list, a

	PITypeException (type_error) is thrown.
charList(String)	Create a Prolog list of one-character atoms from a C#-string.
deList(String)	Create a Prolog list of ASCII codes from a 0-terminated C-string.
npound(String, mV)	Create a compound term with the given name from the given vector of arguments. See PITermV for details.
npound(String, PITerm)	Create a compound term with the given name ant the arguments
npound(String, PITerm, m)	Create a compound term with the given name ant the arguments
npound(String, PITerm, m, PITerm)	Create a compound term with the given name ant the arguments
ng(String)	A SWI-Prolog string represents a byte-string on the global stack. It's

	<p>lifetime is the same as for compound terms and other data living on the global stack. Strings are not only a compound representation of text that is garbage-collected, but as they can contain 0-bytes, they can be used to contain arbitrary C-data structures.</p>
<u>ng(String, Int32)</u>	<p>A SWI-Prolog string represents a byte-string on the global stack. Its lifetime is the same as for compound terms and other data living on the global stack. Strings are not only a compound representation of text that is garbage-collected, but as they can contain 0-bytes, they can be used to contain arbitrary C-data structures.</p>
<u>(PITerm)</u>	<p>PITail is for analysing and constructing lists. It is called PITail as enumeration-steps make the term-reference follow the `tail' of the list.</p> <p>A PITail is created by making a new term-reference pointing to the same object. As PITail is used to enumerate or build a Prolog list, the initial list term-reference keeps pointing to the head of the list.</p>

Type	Get the PIType of a PITerm .
()	Creates a new initialised term (holding a Prolog variable).
to()	Converts to a strongly typed ReadOnlyCollection of PIterm objects that can be accessed by index
ToString()	Converts to a strongly typed Collection of strings that can be accessed by index
ToString()	If PIterm is a list the string is build by calling <code>ToString()</code> for each element in the list separated by ',' and put the brackets around '[' ']'. (Overrides ValueType.ToString() .)
ToStringCanonical()	Convert a PIterm to a string by PL_get_chars/1 with the <code>CVT_WRITE_CANONICAL</code> flag. If it fails <code>PL_get_chars/3</code> is called again with <code>REP_MB</code> flag.

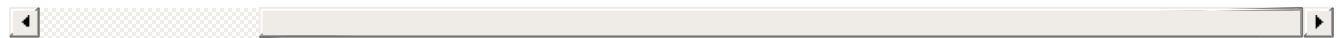
(PITerm)	Unify a PIterm with a PIterm
(String)	Unify a PIterm with a PIterm

S

Mod	Description
	Creates a new initialised term (holding a Prolog variable).
m)	PItail is for analysing and constructing lists.
id(string)	Create compound terms. E.g. by parsing (as read/1) the given text.
'ing)	Create a SWI-Prolog string.
[String)	Create a Prolog list of ASCII codes from a 0-terminated C-string.
String)	Create a Prolog list of one-character atoms from a 0-terminated C-string.

ta, SBS-Softwaresysteme GmbH
ents on this topic to Lesta at SBS-Softwaresysteme.de

SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
))



Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [PITerm\(\)](#)

C# ▾

A new PIterm can be also created by the static methods:

static method	Description
PIVar()	Creates a new initialised term (holding a Prolog variable).
PITail(PITerm)	PITail is for analysing and constructing lists.
PICompound(string)	Create compound terms. E.g. by parsing (as read/1) the given text.
PIString(String)	Create a SWI-Prolog string.
PICodeList(String)	Create a Prolog list of ASCII codes from a 0-terminated C-string.
PICharList(String)	Create a Prolog list of one-character atoms from a 0-terminated C-string.

Members

Icon	Member	Description
≡	PITerm(String)	Creates a term-references holding a Prolog term representing text.
≡	PITerm(Int32)	Creates a term-references holding a Prolog integer representing value.
≡	PITerm(Double)	Creates a term-references holding a Prolog float representing value.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor (value)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [PITerm\(Double\)](#)

C# ▾

Creates a term-references holding a Prolog float representing value.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTerm(
    double value
)

Public Sub New (
    value As Double
)

public:
PlTerm(
    double value
)

new :
    value : float -> PlTerm
```

Parameters

value (Double)

a double value

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta@SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor (value)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► [PlTerm\(Int32\)](#)

C# ▾

Creates a term-references holding a Prolog integer representing value.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTerm(
    int value
)

Public Sub New (
    value As Integer
)

public:
PlTerm(
    int value
)

new :
    value : int -> PlTerm
```

Parameters

value ([Int32](#))

a integer value

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor (text)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [PITerm\(String\)](#)

C#

Creates a term-references holding a Prolog term representing text.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTerm(  
    string text  
)  
  
Public Sub New (  
    text As String  
)  
  
public:  
PlTerm(  
    String^ text  
)  
  
new :  
    text : string -> PlTerm
```

Parameters

text (String)

the text

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta@SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (term)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► [Add\(PlTerm\)](#)

C# ▾

Appends an element to a list by creating a new one and copy all elements Note This is a slow version see my mail from Jan from 2007.11.06 14:44

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool Add(
    PlTerm term
)

Public Function Add (
    term As PlTerm
) As Boolean

public:
bool Add(
    PlTerm term
)

member Add :
    term : PlTerm -> bool
```

Parameters

term ([PlTerm](#))

a closed list

Return Value

[Boolean](#)

True if Succeed

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (listToAppend)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► **AddList(PlTerm)**

C#

Appends a list (PlTail) to a list by creating a new one and copy all elements

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool AddList(
    PlTerm listToAppend
)

Public Function AddList (
    listToAppend As PlTerm
) As Boolean

public:
bool AddList(
    PlTerm listToAppend
)

member AddList :
    listToAppend : PlTerm -> bool
```

Parameters

listToAppend ([PlTerm](#))

a closed list

Return Value

[Boolean](#)

True if Succeed

Examples

```
public void List_Add_list_doc()
{
    PlTerm t = new PlTerm("[x,y]");
    PlTerm l = new PlTerm("[a,b]");

    Assert.IsTrue(l.IsList);
    Assert.IsTrue(l.AddList(t));
    Assert.IsTrue(l.IsList);
    Assert.AreEqual("[a,b,x,y]", l.ToString());
    Assert.AreEqual("a", l.NextValue().ToString());
    Assert.AreEqual("b", l.NextValue().ToString());
    Assert.AreEqual("[x,y]", l.ToString());
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method (term)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Append(PITerm)**

C# ▾

Appends element to the list and make the PITail reference point to the new variable tail. If A is a variable, and this method is called on it using the argument "gnat", a list of the form [gnat|B] is created and the PITail object now points to the new variable B. This method returns TRUE if the unification succeeded and FALSE otherwise. No exceptions are generated.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool Append(
    PITerm term
)

Public Function Append (
    term As PITerm
) As Boolean

public:
bool Append(
    PITerm term
)

member Append :
    term : PITerm -> bool
```

Parameters

term ([PITerm](#))

The PITerm to append on the list.

[-] Return Value

Boolean

true if successful otherwise false

[-] Examples

```
public void List_Append()
{
    PlTerm t = PlTerm.PlVar();
    PlTerm l = PlTerm.PlTail(t);

    Assert.IsTrue(l.Append(new PlTerm("one")));
    Assert.IsTrue(l.Append(new PlTerm("two")));
    Assert.IsTrue(l.Append(new PlTerm("three"))

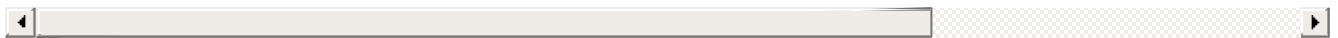
    Assert.AreEqual(1, l.Close());

    Assert.AreEqual(3, list_length(t));
    Assert.AreEqual("[one,two,three]", t.ToString());
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Arity**

Get the arity of the functor if [PITerm](#) is a compound term.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public int Arity { get; }

Public ReadOnly Property Arity As Integer
    Get

public:
property int Arity {
    int get ();
}

member Arity : int with get
```

Property Value

[Int32](#)

Remarks

Arity and [Name](#) are for compound terms only

Exceptions

Exception	Condition
NotSupportedException	Is thrown if the term isn't compound

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Close()**

C#

Unifies the term with [] and returns the result of the unification.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public int Close()
```

```
Public Function Close As Integer
```

```
public:  
int Close()
```

```
member Close : unit -> int
```

Return Value

[Int32](#)

The [int](#) value of [PL_unify_nil\(TermRef\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

CompareTo Method (obj)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ►

CompareTo(Object)

C# ▾

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public int CompareTo(
    Object obj
)

Public Function CompareTo (
    obj As Object
) As Integer

public:
virtual int CompareTo(
    Object^ obj
) sealed

abstract CompareTo :
    obj : Object -> int
override CompareTo :
    obj : Object -> int
```

Parameters

obj ([Object](#))

An object to compare with this instance.

[-] Return Value

[Int32](#)

A 32-bit signed integer that indicates the relative order of the objects being compared. The return value has these meanings:

Value	Meaning
Less than zero	This instance precedes <i>obj</i> in the sort order.
Zero	This instance occurs in the same position in the sort order as <i>obj</i> .
Greater than zero	This instance follows <i>obj</i> in the sort order.

[-] Exceptions

Exception	Condition
<u>ArgumentException</u>	<i>obj</i> is not the same type as this instance.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Equality Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [Equality\(\)](#)

C# ▾

▀ Members

Icon	Member	Description
	Equality(PITerm, PITerm)	Yields TRUE if the PIterm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.
	Equality(PITerm, Int32)	Yields TRUE if the PIterm is an atom or string representing the same integer as the argument, FALSE if the conversion was not successful. conversion of the term is done by PL_get_long
	Equality(Int32, PITerm)	Yields TRUE if the PIterm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.
	Equality(PITerm, String)	Yields TRUE if the PIterm is an atom or string representing the same text as

		the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.
	<u>Equality(String, PIterm)</u>	Yields TRUE if the PIterm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Equality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Equality(PITerm, PITerm)**

C# ▾

Yields TRUE if the PITerm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator ==(  
    PITerm term1,  
    PITerm term2  
)  
  
Public Shared Operator = (  
    term1 As PITerm,  
    term2 As PITerm  
) As Boolean  
  
public:  
static bool operator ==(  
    PITerm term1,  
    PITerm term2  
)  
  
static let inline (=)  
    term1 : PITerm *  
    term2 : PITerm : bool
```

Parameters

term1 ([PITerm](#))

a PIterm

term2 (PITerm)

a PIterm

[-] Return Value

Boolean

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Equality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Equality(PITerm, Int32)**

C# ▾

Yields TRUE if the PIterm is an atom or string representing the same integer as the argument, FALSE if the conversion was not successful. conversion of the term is done by PL_get_long

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator ==(  
    PIterm term,  
    int value  
)  
  
Public Shared Operator = (  
    term As PIterm,  
    value As Integer  
) As Boolean  
  
public:  
static bool operator ==(  
    PIterm term,  
    int value  
)  
  
static let inline (=)  
    term : PIterm *  
    value : int : bool
```

Parameters

***term* (PITerm)**

a PI Term

value ([Int32](#))

a int

Return Value

[Boolean](#)

A bool

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Equality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Equality(PITerm, String)**

C# ▾

Yields TRUE if the PIterm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator ==(  
    PIterm term,  
    string value  
)  
  
Public Shared Operator = (  
    term As PIterm,  
    value As String  
) As Boolean  
  
public:  
static bool operator ==(  
    PIterm term,  
    String^ value  
)  
  
static let inline (=)  
    term : PIterm *  
    value : string : bool
```

Parameters

***term* (PITerm)**

[Missing <param name="term"/> documentation for
"M:SbsSW.SwiPICs.PITerm.op_Equality(SbsSW.SwiPICs.PITer

value ([String](#))

[Missing <param name="value"/> documentation for
"M:SbsSW.SwiPICs.PITerm.op_Equality(SbsSW.SwiPICs.PITer

▀ Return Value

[Boolean](#)

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Equality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Equality(Int32, PITerm)**

C# ▾

Yields TRUE if the PITerm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the string are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator ==(  
    int value,  
    PITerm term  
)  
  
Public Shared Operator = (  
    value As Integer,  
    term As PITerm  
) As Boolean  
  
public:  
static bool operator ==(  
    int value,  
    PITerm term  
)  
  
static let inline (=)  
    value : int *  
    term : PITerm : bool
```

Parameters

value ([Int32](#))

[Missing <param name="value"/> documentation for
"M:SbsSW.SwiPICs.PIterm.op_Equality(System.Int32,SbsSW.S
term ([PIterm](#))

[Missing <param name="term"/> documentation for
"M:SbsSW.SwiPICs.PIterm.op_Equality(System.Int32,SbsSW.S

☒ Return Value

[Boolean](#)

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Equality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Equality(String, PITerm)**

C# ▾

Yields TRUE if the PITerm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator ==(  
    string value,  
    PITerm term  
)  
  
Public Shared Operator = (  
    value As String,  
    term As PITerm  
) As Boolean  
  
public:  
static bool operator ==(  
    String^ value,  
    PITerm term  
)  
  
static let inline (=)  
    value : string *  
    term : PITerm : bool
```

Parameters

value (String)

[Missing <param name="value"/> documentation for
"M:SbsSW.SwiPICs.PIterm.op_Equality(System.String,SbsSW

term ([PIterm](#))

[Missing <param name="term"/> documentation for
"M:SbsSW.SwiPICs.PIterm.op_Equality(System.String,SbsSW

❑ Return Value

[Boolean](#)

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method (obj)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Equals(Object)**

C# ▾

Indicates whether this instance and a specified object are equal.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override bool Equals(
    Object obj
)
```

```
Public Overrides Function Equals (
    obj As Object
) As Boolean
```

```
public:
virtual bool Equals(
    Object^ obj
) override
```

```
abstract Equals :
    obj : Object -> bool
override Equals :
    obj : Object -> bool
```

Parameters

obj ([Object](#))

Another object to compare to.

Return Value

[Boolean](#)

true if *obj* and this instance are the same type and represent the same

value; otherwise, false.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog **Explicit**
Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Explicit()**

C# ▾

▀ **Members**

Icon	Member	Description
	Explicit(PITerm to String)	Converts the Prolog argument into a string which implies Prolog atoms and strings are converted to the represented text or throw a PITypeException.
	Explicit(PITerm to Int32)	Yields a int if the PITerm is a Prolog integer or float that can be converted without loss to a int. Throws a PITypeException exception otherwise
	Explicit(PITerm to Double)	Yields the value as a C# double if PITerm represents a Prolog integer or float. Throws a PITypeException exception otherwise.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Explicit Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Explicit(PITerm to Double)**

C# ▾

Yields the value as a C# double if PIterm represents a Prolog integer or float. Throws a PITypeException exception otherwise.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static explicit operator double (
    PIterm term
)

Public Shared Narrowing Operator CType (
    term As PIterm
) As Double

static explicit operator double (
    PIterm term
)

F# does not support the declaration of new ca
```

Parameters

term ([PITerm](#))

A PIterm represents a Prolog integer or float

Return Value

[Double](#)

A C# double

Exceptions

Exception	Condition
PITypeException	Throws a PITypeException exception if PIType is not a PIType.PInteger or a PIType.PIFloat.
PreconditionException	Is thrown if the operator is used on an uninitialized PITerm

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Explicit Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Explicit(PITerm to String)**

C# ▾

Converts the Prolog argument into a string which implies Prolog atoms and strings are converted to the represented text or throw a PITypeException.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static explicit operator string (
    PITerm term
)

Public Shared Narrowing Operator CType (
    term As PITerm
) As String

static explicit operator String^ (
    PITerm term
)

F# does not support the declaration of new ca
```

Parameters

term ([PITerm](#))

A PITerm that can be converted to a string

Return Value

[String](#)

A C# string

Remarks

Converts the Prolog argument using PL_get_chars() using the flags CVT_ALL|CVT_WRITE|BUF_RING, which implies Prolog atoms and strings are converted to the represented text or throw a PITypeException.

If the above call return 0 [PL_get_chars](#) is called a second time with the flags CVT_ALL|CVT_WRITE|BUF_RING|REP_UTF8.

All other data is handed to write/1.

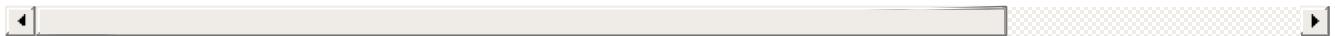
Exceptions

Exception	Condition
PITypeException	Throws a PITypeException exception
PreconditionException	Is thrown if the operator is used on an uninitialized PITerm

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Explicit Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Explicit(PITerm to Int32)**

C# ▾

Yields a int if the PIterm is a Prolog integer or float that can be converted without loss to a int. Throws a PITypeException exception otherwise

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static explicit operator int (
    PIterm term
)
```

```
Public Shared Narrowing Operator CType (
    term As PIterm
) As Integer
```

```
static explicit operator int (
    PIterm term
)
```

F# does not support the declaration of new operators.

Parameters

term ([PITerm](#))

A PIterm is a Prolog integer or float that can be converted without loss to a int.

Return Value

[Int32](#)

A C# int

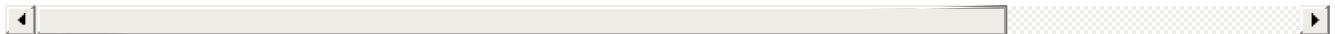
Exceptions

Exception	Condition
PITypeException	Throws a PITypeException exception if PIType is not a PIType.PIInteger or a PIType.PIFloat.
PreconditionException	Is thrown if the operator is used on an uninitialized PITerm

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta%20at%20SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **GetEnumerator Method**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [GetEnumerator\(\)](#)

C#

Returns an enumerator that iterates through the collection.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public IEnumarator<PTerm> GetEnumerator()  
  
Public Function GetEnumerator As IEnumarator()  
  
public:  
    virtual IEnumarator<PTerm>^ GetEnumerator()  
  
abstract GetEnumerator : unit -> IEnumarator<  
override GetEnumerator : unit -> IEnumarator<
```

Return Value

[IEnumarator<PTerm>](#)

A System.Collections.Generic.IEnumarator<T> that can be used to iterate through the collection.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **GetHashCode Method**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **GetHashCode()**

C# ▾

Returns the hash code for this instance.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override int GetHashCode()  
  
Public Overrides Function GetHashCode As Integer  
  
public:  
    virtual int GetHashCode() override  
  
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

[Int32](#)

A 32-bit signed integer that is the hash code for this instance.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **GreaterThan Operator**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ►

GreaterThan(PITerm, PITerm)

C# ▾

Yields TRUE if the PITerm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator >(
    PITerm term1,
    PITerm term2
)

Public Shared Operator > (
    term1 As PITerm,
    term2 As PITerm
) As Boolean

public:
static bool operator >(
    PITerm term1,
    PITerm term2
)

static let inline (>)
    term1 : PITerm *
    term2 : PITerm : bool
```

Parameters

term1 ([PITerm](#))

a PIterm

term2 (PITerm)

a PIterm

[-] Return Value

Boolean

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog GreaterThanOrEqual Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ►

GreaterThanOrEqual(PITerm, PITerm)

C# ▾

Yields TRUE if the PITerm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator >=
    PITerm term1,
    PITerm term2
)

Public Shared Operator >= (
    term1 As PITerm,
    term2 As PITerm
) As Boolean

public:
static bool operator >=
    PITerm term1,
    PITerm term2
)

static let inline (>=)
    term1 : PITerm *
    term2 : PITerm : bool
```

Parameters

term1 ([PITerm](#))

a PIterm

term2 (PITerm)

a PIterm

[-] Return Value

Boolean

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Inequality Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [Inequality\(\)](#)

C# ▾

Inequality Method overload

[Equality\(PITerm, PITerm\)](#) a [Equality\(PITerm, Int32\)](#)

▀ Members

Icon	Member	Description
	Inequality(PITerm, PITerm)	Yields TRUE if the PIterm is an atom or string representing the same text as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.
	Inequality(PITerm, Int32)	summary
	Inequality(Int32, PITerm)	summary
	Inequality(PITerm, String)	test
	Inequality(String, PITerm)	test

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Inequality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Inequality(PITerm, PITerm)**

C# ▾

Yields TRUE if the PITerm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator !=(
    PITerm term1,
    PITerm term2
)

Public Shared Operator <> (
    term1 As PITerm,
    term2 As PITerm
) As Boolean

public:
static bool operator !=(
    PITerm term1,
    PITerm term2
)

static let inline (<>)
    term1 : PITerm *
    term2 : PITerm : bool
```

Parameters

term1 ([PITerm](#))

a PIterm

term2 (PITerm)

a PIterm

[-] Return Value

Boolean

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Inequality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Inequality(PITerm, Int32)**

C# ▾

summary

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator !=(
    PITerm term,
    int value
)

Public Shared Operator <> (
    term As PITerm,
    value As Integer
) As Boolean

public:
static bool operator !=(
    PITerm term,
    int value
)

static let inline (<>)
    term : PITerm *
    value : int  : bool
```

Parameters

term ([PITerm](#))

[Missing <param name="term"/> documentation for

"M:SbsSW.SwiPICs.PIterm.op_Inequality(SbsSW.SwiPICs.PIterm, SbsSW.SwiPICs.PIterm, value)"

value (Int32)

[Missing <param name="value"/> documentation for

"M:SbsSW.SwiPICs.PIterm.op_Inequality(SbsSW.SwiPICs.PIterm, SbsSW.SwiPICs.PIterm, value)"

▀ Return Value

Boolean

[Missing <returns> documentation for

"M:SbsSW.SwiPICs.PIterm.op_Inequality(SbsSW.SwiPICs.PIterm, SbsSW.SwiPICs.PIterm, value)"

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Inequality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Inequality(PITerm, String)**

C# ▾

test

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator !=(
    PITerm term,
    string value
)

Public Shared Operator <> (
    term As PITerm,
    value As String
) As Boolean

public:
static bool operator !=(
    PITerm term,
    String^ value
)

static let inline (<>)
    term : PITerm *
    value : string : bool
```

Parameters

term ([PITerm](#))

[Missing <param name="term"/> documentation for

"M:SbsSW.SwiPICs.PIterm.op_Inequality(SbsSW.SwiPICs.PIterm, SbsSW.SwiPICs.PIterm, value, result)"

value ([String](#))

[Missing <param name="value"/> documentation for

"M:SbsSW.SwiPICs.PIterm.op_Inequality(SbsSW.SwiPICs.PIterm, SbsSW.SwiPICs.PIterm, value, result)"

Return Value

[Boolean](#)

[Missing <returns> documentation for

"M:SbsSW.SwiPICs.PIterm.op_Inequality(SbsSW.SwiPICs.PIterm, SbsSW.SwiPICs.PIterm, value, result)"

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Inequality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Inequality(Int32, PITerm)**

C# ▾

summary

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator !=(
    int value,
    PITerm term
)

Public Shared Operator <> (
    value As Integer,
    term As PITerm
) As Boolean

public:
static bool operator !=(
    int value,
    PITerm term
)

static let inline (<>)
    value : int *
    term : PITerm : bool
```

Parameters

value (Int32)

[Missing <param name="value"/> documentation for

"M:SbsSW.SwiPICs.PIterm.op_Inequality(System.Int32,SbsSW

term ([PIterm](#))

[Missing <param name="term"/> documentation for
"M:SbsSW.SwiPICs.PIterm.op_Inequality(System.Int32,SbsSW

▀ Return Value

[Boolean](#)

[Missing <returns> documentation for
"M:SbsSW.SwiPICs.PIterm.op_Inequality(System.Int32,SbsSW.Sw

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Inequality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Inequality(String, PITerm)**

C# ▾

test

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator !=(
    string value,
    PITerm term
)

Public Shared Operator <> (
    value As String,
    term As PITerm
) As Boolean

public:
static bool operator !=(
    String^ value,
    PITerm term
)

static let inline (<>)
    value : string *
    term : PITerm : bool
```

Parameters

value (String)

[Missing <param name="value"/> documentation for

"M:SbsSW.SwiPICs.PIterm.op_Inequality(System.String,SbsS"

term ([PIterm](#))

[Missing <param name="term"/> documentation for
"M:SbsSW.SwiPICs.PIterm.op_Inequality(System.String,SbsS"

▀ Return Value

[Boolean](#)

[Missing <returns> documentation for
"M:SbsSW.SwiPICs.PIterm.op_Inequality(System.String,SbsSW.Sv

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsAtom**

C# ▾

Return true if [PITerm](#) is an atom.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsAtom { get; }

Public ReadOnly Property IsAtom As Boolean
    Get

public:
property bool IsAtom {
    bool get ();
}

member IsAtom : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsAtomic**

C#

Return true if [PITerm](#) is atomic (not variable or compound).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsAtomic { get; }

Public ReadOnly Property IsAtomic As Boolean
    Get

public:
property bool IsAtomic {
    bool get ();
}

member IsAtomic : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

IsCompound Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsCompound**

C# ▾

Return true if [PITerm](#) is a compound term. Note that a list is a compound term ./2

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsCompound { get; }

Public ReadOnly Property IsCompound As Boolean
    Get

public:
property bool IsCompound {
    bool get ();
}

member IsCompound : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsFloat**

C# ▾

Return true if [PITerm](#) is a float.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsFloat { get; }

Public ReadOnly Property IsFloat As Boolean
    Get

public:
property bool IsFloat {
    bool get ();
}

member IsFloat : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsGround**

C# ▾

Return true if [PITerm](#) is a ground term. See also [ground/1](#). This function is cycle-safe.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsGround { get; }

Public ReadOnly Property IsGround As Boolean
    Get

public:
property bool IsGround {
    bool get ();
}

member IsGround : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **IsInitialized Property**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsInitialized**

C#

return false for a PIterm variable which is only declared and true if it is also initialized

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsInitialized { get; }

Public ReadOnly Property IsInitialized As Boolean
    Get

public:
property bool IsInitialized {
    bool get ();
}

member IsInitialized : bool with get
```

Property Value

[Boolean](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsInteger**

C# ▾

Return true if [PITerm](#) is an integer.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsInteger { get; }

Public ReadOnly Property IsInteger As Boolean
    Get

public:
property bool IsInteger {
    bool get ();
}

member IsInteger : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsList**

C# ▾

Return true if [PITerm](#) is a compound term with functor ./2 or the atom [].

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsList { get; }

Public ReadOnly Property IsList As Boolean
    Get

public:
property bool IsList {
    bool get ();
}

member IsList : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsNumber**

C# ▾

Return true if [PITerm](#) is an integer or float.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsNumber { get; }

Public ReadOnly Property IsNumber As Boolean
    Get

public:
property bool IsNumber {
    bool get ();
}

member IsNumber : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsString**

C# ▾

Return true if [PITerm](#) is a string.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsString { get; }

Public ReadOnly Property IsString As Boolean
    Get

public:
property bool IsString {
    bool get ();
}

member IsString : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **IsVar**

Return true if [PITerm](#) is a variable

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool IsVar { get; }

Public ReadOnly Property IsVar As Boolean
    Get

public:
property bool IsVar {
    bool get ();
}

member IsVar : bool with get
```

Property Value

[Boolean](#)

See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property (position)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► [Item\[Int32\]](#)

C# ▾

If PlTerm is compound and index is between 0 and Arity (including), the nth PlTerm is returned.

If pos is 0 the functor of the term is returned (For a list '.').

See: [PL_get_arg/3](#)

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTerm this[
    int position
] { get; }

Public ReadOnly Default Property Item (
    position As Integer
) As PlTerm
    Get

public:
property PlTerm default[int position] {
    PlTerm get (int position);
}

member Item : PlTerm with get
```

Parameters

position ([Int32](#))

To Get the nth PlTerm

▀ Return Value

[PITerm](#)

a PIterm

▀ Examples

[Copy](#)

```
[TestMethod]
public void PItermIndexer2()
{
    var t = new PIterm("foo(bar, x(y))");
    Assert.AreEqual("foo", t[0].ToString());
    Assert.AreEqual("bar", t[1].ToString());
    Assert.AreEqual("x(y)", t[2].ToString());
}
```

```
[TestMethod]
public void PItermIndexer_list1()
{
    var t = new PIterm("[a,b,c]");
    Assert.AreEqual(".", t[0].ToString());
    Assert.AreEqual("a", t[1].ToString());
    Assert.AreEqual("[b,c]", t[2].ToString())
}
```

▀ Exceptions

Exception	Condition
NotSupportedException	Is thrown if PIterm is not of Type PICompound see IsCompound
ArgumentOutOfRangeException	Is thrown if (pos < 0 pos >= Arity)
InvalidOperationException	Is thrown if PL_get_arg returns 0.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

LessThan Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **LessThan(PITerm, PITerm)**

C# ▾

Yields TRUE if the PITerm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator <(
    PITerm term1,
    PITerm term2
)

Public Shared Operator < (
    term1 As PITerm,
    term2 As PITerm
) As Boolean

public:
static bool operator <(
    PITerm term1,
    PITerm term2
)

static let inline (<)
    term1 : PITerm *
    term2 : PITerm : bool
```

Parameters

term1 ([PITerm](#))

a PIterm

term2 (PITerm)

a PIterm

[-] Return Value

Boolean

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **LessThanOrEqual Operator**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ►

LessThanOrEqual(PITerm, PITerm)

C# ▾

Yields TRUE if the PITerm is an atom or string representing the same term as the argument, FALSE if the conversion was successful, but the strings are not equal and an type_error exception if the conversion failed.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator <=
    PITerm term1,
    PITerm term2
)

Public Shared Operator <= (
    term1 As PITerm,
    term2 As PITerm
) As Boolean

public:
static bool operator <=
    PITerm term1,
    PITerm term2
)

static let inline (<=)
    term1 : PITerm *
    term2 : PITerm : bool
```

Parameters

term1 ([PITerm](#))

a PI Term

term2 (PITerm)

a PI Term

[-] Return Value

Boolean

true or false

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Name**

C# ▾

Get a holding the name of the functor if [PITerm](#) is a compound term.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public string Name { get; }

Public ReadOnly Property Name As String
    Get

public:
property String^ Name {
    String^ get ();
}

member Name : string with get
```

Property Value

[String](#)

Remarks

[Arity](#) and **Name** are for compound terms only

Exceptions

Exception	Condition
NotSupportedException	Is thrown if the term isn't compound

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **NextValue** Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [NextValue\(\)](#)

C# ▾

return a PIterm bound to the next element of the list PItail and advance PItail. Returns the element on success or a free PIterm (Variable) if PItail represents the empty list. If PItail is neither a list nor the empty lis a PITypeException (type_error) is thrown.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIterm NextValue()

Public Function NextValue As PIterm

public:
PIterm NextValue()

member NextValue : unit -> PIterm
```

Return Value

[PITerm](#)

The Next element in the list as a PIterm which is a variable for the last element or an empty list

Examples

```
public void List_Add_list_doc()
{
    PIterm t = new PIterm("[x,y]");
    PIterm l = new PIterm("[a,b]");

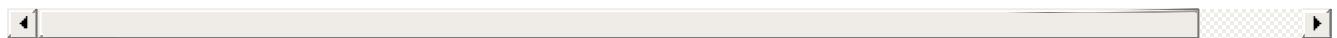
    Assert.IsTrue(l.IsList);
```

```
    Assert.IsTrue(l.AddList(t));
    Assert.IsTrue(l.IsList);
    Assert.AreEqual("[a,b,x,y]", l.ToString());
    Assert.AreEqual("a", l.NextValue().ToString());
    Assert.AreEqual("b", l.NextValue().ToString());
    Assert.AreEqual("[x,y]", l.ToString());
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PlCharList Method (text)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► [PlCharList\(String\)](#)

C#

Create a Prolog list of one-character atoms from a C#-string.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PlTerm PlCharList(
    string text
)

Public Shared Function PlCharList (
    text As String
) As PlTerm

public:
static PlTerm PlCharList(
    String^ text
)

static member PlCharList :
    text : string -> PlTerm
```

Parameters

text (String)

a string

Return Value

[PlTerm](#)

A new PlTerm containing a prolog list of character

Remarks

Character lists are compliant to Prolog's [atom_chars/2](#) predicate.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PICodeList Method (text)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [PICodeList\(String\)](#)

C#

Create a Prolog list of ASCII codes from a 0-terminated C-string.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PIterm PICodeList(
    string text
)

Public Shared Function PICodeList (
    text As String
) As PIterm

public:
static PIterm PICodeList(
    String^ text
)

static member PICodeList :
    text : string -> PIterm
```

Parameters

text ([String](#))

The text

Return Value

[PITerm](#)

a new [PITerm](#)

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PICompound Method**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [PICompound\(\)](#)

C# ▾

▀ Members

Icon	Member	Description
= 	PICompound(String, PITermV)	Create a compound term with the given name from the given vector of arguments. See PITermV for details.
= 	PICompound(String, PITerm)	Create a compound term with the given name ant the arguments
= 	PICompound(String, PITerm, PITerm)	Create a compound term with the given name ant the arguments
= 	PICompound(String, PITerm, PITerm, PITerm)	Create a compound term with the given name ant the arguments

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PICompound Method (functor, arg1)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ►

PICompound(String, PIterm)

C# ▾

Create a compound term with the given name ant the arguments

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PIterm PICompound(
    string functor,
    PIterm arg1
)
```

```
Public Shared Function PICompound (
    functor As String,
    arg1 As PIterm
) As PIterm
```

```
public:
static PIterm PICompound(
    String^ functor,
    PIterm arg1
)
```

```
static member PICompound :
    functor : string *
    arg1 : PIterm -> PIterm
```

Parameters

***functor* ([String](#))**

The functor (name) of the compound term

***arg1* ([PITerm](#))**

The first Argument as a [PITerm](#)

▀ Return Value

[PITerm](#)

a new [PITerm](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog PICompound Method (functor, arg1, arg2)[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ►[PICompound\(String, PIterm, PIterm\)](#)

C# ▾

Create a compound term with the given name ant the arguments

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PIterm PICompound(
    string functor,
    PIterm arg1,
    PIterm arg2
)
```

```
Public Shared Function PICompound (
    functor As String,
    arg1 As PIterm,
    arg2 As PIterm
) As PIterm
```

```
public:
static PIterm PICompound(
    String^ functor,
    PIterm arg1,
    PIterm arg2
)
```

```
static member PICompound :
    functor : string *
    arg1 : PIterm *
    arg2 : PIterm -> PIterm
```

[-] Parameters

***functor* ([String](#))**

The functor (name) of the compound term

***arg1* ([PITerm](#))**

The first Argument as a [PITerm](#)

***arg2* ([PITerm](#))**

The second Argument as a [PITerm](#)

[-] Return Value

[PITerm](#)

a new [PITerm](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PICompound Method (functor, arg1, arg2, arg3)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ►
PICompound(String, PIterm, PIterm, PIterm)

C# ▾

Create a compound term with the given name ant the arguments

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PIterm PICompound(
    string functor,
    PIterm arg1,
    PIterm arg2,
    PIterm arg3
)
```

```
Public Shared Function PICompound (
    functor As String,
    arg1 As PIterm,
    arg2 As PIterm,
    arg3 As PIterm
) As PIterm
```

```
public:
static PIterm PICompound(
    String^ functor,
    PIterm arg1,
    PIterm arg2,
    PIterm arg3
)
```

```
static member PlCompound :  
    functor : string *  
    arg1 : PlTerm *  
    arg2 : PlTerm *  
    arg3 : PlTerm -> PlTerm
```

■ Parameters

functor ([String](#))

The functor (name) of the compound term

arg1 ([PlTerm](#))

The first Argument as a [PlTerm](#)

arg2 ([PlTerm](#))

The second Argument as a [PlTerm](#)

arg3 ([PlTerm](#))

The third Argument as a [PlTerm](#)

■ Return Value

[PlTerm](#)

a new [PlTerm](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PlCompound Method (functor, args)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ►

PlCompound(String, PlTermV)

C# ▾

Create a compound term with the given name from the given vector of arguments. See [PlTermV](#) for details.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PlTerm PlCompound(
    string functor,
    PlTermV args
)

Public Shared Function PlCompound (
    functor As String,
    args As PlTermV
) As PlTerm

public:
static PlTerm PlCompound(
    String^ functor,
    PlTermV args
)

static member PlCompound :
    functor : string *
    args : PlTermV -> PlTerm
```

Parameters

***functor* ([String](#))**

The functor (name) of the compound term

args ([PlTermV](#))

the arguments as a [PlTermV](#)

[-] Return Value

[PlTerm](#)

a new [PlTerm](#)

[-] Examples

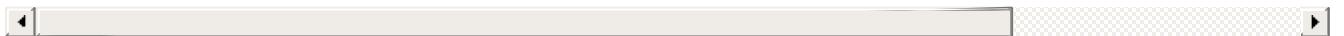
The example below creates the Prolog term hello(world).

```
PlTerm t = PlTerm.PlCompound("hello", new PlT
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta%20at%20SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **PIString()**

C# ▾

▀ Members

Icon	Member	Description
	PIString(String)	A SWI-Prolog string represents a byte string on the global stack. Its lifetime is the same as for compound terms and other data living on the global stack. Strings are not only a compound representation of text that is garbage-collected, but as they can contain 0-bytes, they can be used to contain arbitrary C-data structures.
	PIString(String, Int32)	A SWI-Prolog string represents a byte string on the global stack. Its lifetime is the same as for compound terms and other data living on the global stack. Strings are not only a compound representation of text that is garbage-collected, but as they can contain 0-bytes, they can be used to contain arbitrary C-data structures.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (text)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► **PlString(String)**

C# ▾

A SWI-Prolog string represents a byte-string on the global stack. Its lifetime is the same as for compound terms and other data living on the global stack. Strings are not only a compound representation of text that is garbage-collected, but as they can contain 0-bytes, they can be used to contain arbitrary C-data structures.

- Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PlTerm PlString(
    string text
)

Public Shared Function PlString (
    text As String
) As PlTerm

public:
static PlTerm PlString(
    String^ text
)

static member PlString :
    text : string -> PlTerm
```

- Parameters

text (String)

the string

- Return Value

PITerm

a new PIterm

■ Remarks

NOTE: this Method do *not* work with unicode characters. Consider to use new PIterm(test) instead.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (text, len)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► **PlString(String, Int32)**

C# ▾

A SWI-Prolog string represents a byte-string on the global stack. Its lifetime is the same as for compound terms and other data living on the global stack. Strings are not only a compound representation of text that is garbage-collected, but as they can contain 0-bytes, they can be used to contain arbitrary C-data structures.

- Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PlTerm PlString(
    string text,
    int len
)

Public Shared Function PlString (
    text As String,
    len As Integer
) As PlTerm

public:
static PlTerm PlString(
    String^ text,
    int len
)

static member PlString :
    text : string *
    len : int -> PlTerm
```

- Parameters

text (String)

the string

len (Int32)

the length of the string

- Return Value

[PITerm](#)

a new PIterm

- Remarks

NOTE: this Method do *not* work with unicode characters. Consider to use new PIterm(test) instead.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta%20at%20SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (list)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► [PlTail\(PlTerm\)](#)

C# ▾

PlTail is for analysing and constructing lists. It is called PlTail as enumeration-steps make the term-reference follow the 'tail' of the list.

A PlTail is created by making a new term-reference pointing to the same object. As PlTail is used to enumerate or build a Prolog list, the initial list term-reference keeps pointing to the head of the list.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PlTerm PlTail(
    PlTerm list
)

Public Shared Function PlTail (
    list As PlTerm
) As PlTerm

public:
static PlTerm PlTail(
    PlTerm list
)

static member PlTail :
    list : PlTerm -> PlTerm
```

Parameters

list ([PlTerm](#))

The initial PlTerm

■ Return Value

[PlTerm](#)

A PlTerm for which is_list/1 succeed.

■ Examples

```
public void List_Append()
{
    PlTerm t = PlTerm.PlVar();
    PlTerm l = PlTerm.PlTail(t);

    Assert.IsTrue(l.Append(new PlTerm("one")));
    Assert.IsTrue(l.Append(new PlTerm("two")));
    Assert.IsTrue(l.Append(new PlTerm("three"))

    Assert.AreEqual(1, l.Close());

    Assert.AreEqual(3, list_length(t));
    Assert.AreEqual("[one,two,three]", t.ToString());
}
```

■ See Also

[Append\(PlTerm\)](#)

[Add\(PlTerm\)](#)

[AddList\(PlTerm\)](#)

[Close\(\)](#)

[NextValue\(\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► [PlType](#)

C# ▾

Get the [PlType](#) of a [PlTerm](#).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlType PlType { get; }

Public ReadOnly Property PlType As PlType
    Get

public:
property PlType PlType {
    PlType get ();
}

member PlType : PlType with get
```

Property Value

[PlType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► **PlVar()**

C# ▾

Creates a new initialised term (holding a Prolog variable).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static PlTerm PlVar()  
  
Public Shared Function PlVar As PlTerm  
  
public:  
    static PlTerm PlVar()  
  
    static member PlVar : unit -> PlTerm
```

Return Value

[PlTerm](#)

a PlTerm

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **ToList()**

C# ▾

Converts to a strongly typed ReadOnlyCollection of PIterm objects that can be accessed by index

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public ReadOnlyCollection<PITerm> ToList()

Public Function ToList As ReadOnlyCollection<
public:
    ReadOnlyCollection<PITerm>^ ToList()

member ToList : unit -> ReadOnlyCollection<PI
```

Return Value

[ReadOnlyCollection<PITerm>](#)

A strongly typed ReadOnlyCollection of PIterm objects

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **ToListString Method**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [ToListString\(\)](#)

C#

Converts to a strongly typed Collection of strings that can be accessed by index

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public Collection<string> ToListString()  
  
Public Function ToListString As Collection(Of  
public:  
Collection<String^>^ ToListString()  
  
member ToListString : unit -> Collection<stri
```

Return Value

[Collection<String>](#)

A strongly typed string Collection

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► [ToString\(\)](#)

C# ▾

If PIterm is a list the string is build by calling ToString() for each element in the list separated by ',' and put the brackets around '[' ']'.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override string ToString()

Public Overrides Function ToString As String

public:
virtual String^ ToString() override

abstract ToString : unit -> string
override ToString : unit -> string
```

Return Value

[String](#)

A string representing the PIterm.

See Also

[\[Overload:string\]](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **ToStringCanonical** Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ►

ToStringCanonical()

C# ▾

Convert a PIterm to a string by [PL_get_chars/1](#) with the CVT_WRITE_CANONICAL flag. If it fails PL_get_chars/3 is called again with REP_MB flag.

[-] Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public string ToStringCanonical()

Public Function ToStringCanonical As String

public:
String^ ToStringCanonical()

member ToStringCanonical : unit -> string
```

[-] Return Value

[String](#)

return the string of a PIterm

[-] Exceptions

Exception	Condition
PITypeException	Throws a PITypeException if PL_get_chars/3 didn't succeeds.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Unify()**

C# ▾

This methods performs Prolog unification and returns true if successful and false otherwise. It is equal to the prolog =/2 operator.

See [Unify\(PITerm\)](#) for an example.

Remarks

This methods are introduced for clear separation between the destructive assignment in C# using = and prolog unification.

Members

Icon	Member	Description
=	Unify(PITerm)	Unify a PIterm with a PIterm
=	Unify(String)	Unify a PIterm with a PIterm

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (term)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTerm](#) ► [Unify\(PlTerm\)](#)

C# ▾

Unify a PlTerm with a PlTerm

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool Unify(
    PlTerm term
)

Public Function Unify (
    term As PlTerm
) As Boolean

public:
bool Unify(
    PlTerm term
)

member Unify :
    term : PlTerm -> bool
```

Parameters

term ([PlTerm](#))

the second term for unification

Return Value

[Boolean](#)

true or false

Examples

Cop

```
public void UnifyTermVar_doc()
{
    var t1 = new PlTerm("x(A, 2)");
    var t2 = new PlTerm("x(1, B)");
    Assert.IsTrue(t1.Unify(t2));
    Assert.AreEqual("x(1,2)", t1.ToString());
    Assert.AreEqual("x(1,2)", t2.ToString());
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (atom)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITerm](#) ► **Unify(String)**

C# ▾

Unify a PIterm with a PIterm

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool Unify(
    string atom
)

Public Function Unify (
    atom As String
) As Boolean

public:
bool Unify(
    String^ atom
)

member Unify :
    atom : string -> bool
```

Parameters

atom ([String](#))

A string to unify with

Return Value

[Boolean](#)

true or false

Examples

Cop

```
public void UnifyTermVar_doc()
{
    var t1 = new PlTerm("x(A, 2)");
    var t2 = new PlTerm("x(1, B)");
    Assert.IsTrue(t1.Unify(t2));
    Assert.AreEqual("x(1,2)", t1.ToString());
    Assert.AreEqual("x(1,2)", t2.ToString());
}
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Structure

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► **PITermV**

C# ▾

The struct PITermv represents an array of term-references

This type is used to pass the arguments to a foreign defined predicate (see [DelegateParameterVarArgs](#)), construct compound terms (see [PICompound\(String, PITermV\)](#)) and to create queries (see [PILQuery](#)).

The only useful member function is the overloading of [], providing (0-based) access to the elements. [Item\[Int32\]](#) Range checking is performed and raises a ArgumentOutOfRangeException exception.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public struct PITermV : IEquatable<PITermV>

Public Structure PITermV
    Implements IEquatable(Of PITermV)

public value class PITermV : IEquatable<PITer
[<SealedAttribute>]
type PITermV =
    struct
        interface IEquatable<PITermV>
    end
```

Members

All Members

Constructors

Methods

Properties

Public

Instance



Declare

<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Static	<input checked="" type="checkbox"/> Inherited
Icon	Member	Description
	PITermV(Int32)	Create a vector of PI Terms with <i>size</i> elements
	PITermV(PITerm)	Create a PI Term V from the given PI Terms.
	PITermV(PITerm, PITerm)	Create a PI Term V from the given PI Terms.
	PITermV(PITerm, PITerm, PITerm)	Create a PI Term V from the given PI Terms.
	PITermV(PITerm[])	Create a PI Term V from the given PI Term[] array.
	Equality(PITermV, PITermV)	
	Equals(Object)	Indicates whether this instance and a specified object are equal. (Overrides ValueType.Equals(Object))
	Equals(PITermV)	Compare the size and A0 of the PI term V
	Finalize()	Allows an object to try to free

		resources and perform other cleanup operations before it is reclaimed by garbage collection.
		(Inherited from Object .)
	GetHashCode()	Returns the hash code for this instance. (Overrides ValueType.GetHashCode() .)
	GetType()	Gets the type of the current instance. (Inherited from Object .)
	Inequality(PITermV, PITermV)	
	Item[Int32]	A zero based list
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Size	Get the size of a PITermV
	ToString()	Returns the fully qualified type name of this instance. (Inherited from ValueType .)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0

(1.1.60605.0)



Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► [PITermV\(\)](#)

C#

Create a PItermV vector from the given PIterm parameters

Create a new vector with PIterm as elements

It can be created with *size* elements

or

automatically for 1, 2 or 3 pITerms

▀ Members

Icon	Member	Description
	PITermV(Int32)	Create a vector of PIterms with <i>size</i> elements
	PITermV(PIterm)	Create a PItermV from the given PIterms.
	PITermV(PIterm, PIterm)	Create a PItermV from the given PIterms.
	PITermV(PIterm, PIterm, PIterm)	Create a PItermV from the given PIterms.
	PITermV(PIterm[])	Create a PItermV from the given PIterm[] array.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor (term0)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► [PITermV\(PITerm\)](#)

C# ▾

Create a PItermV from the given [PITerms](#).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PItermV(
    PIterm term0
)

Public Sub New (
    term0 As PIterm
)

public:
PItermV(
    PIterm term0
)

new :
    term0 : PIterm -> PItermV
```

Parameters

term0 ([PITerm](#))

The first [PITerm](#) in the vector.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor (term0, term1)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► [PITermV\(PITerm, PITerm\)](#)

C# ▾

Create a PITermV from the given [PITerms](#).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PITermV(  
    PITerm term0,  
    PITerm term1  
)  
  
Public Sub New (  
    term0 As PITerm,  
    term1 As PITerm  
)  
  
public:  
PITermV(  
    PITerm term0,  
    PITerm term1  
)  
  
new :  
    term0 : PITerm *  
    term1 : PITerm -> PITermV
```

Parameters

term0 ([PITerm](#))

The first [PITerm](#) in the vector.

***term1* ([PITerm](#))**

The second [PITerm](#) in the vector.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog **PITermV**

Constructor (term0, term1, term2)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► [PITermV\(PITerm, PITerm, PITerm\)](#)

C# ▾

Create a PITermV from the given [PITerms](#).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PITermV(  
    PITerm term0,  
    PITerm term1,  
    PITerm term2  
)  
  
Public Sub New (  
    term0 As PITerm,  
    term1 As PITerm,  
    term2 As PITerm  
)  
  
public:  
PITermV(  
    PITerm term0,  
    PITerm term1,  
    PITerm term2  
)  
  
new :  
    term0 : PITerm *  
    term1 : PITerm *  
    term2 : PITerm -> PITermV
```

■ Parameters

***term0* ([PITerm](#))**

The first [PITerm](#) in the vector.

***term1* ([PITerm](#))**

The second [PITerm](#) in the vector.

***term2* ([PITerm](#))**

The third [PITerm](#) in the vector.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Constructor (terms)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► [PITermV\(PITerm\[\]\)](#)

C# ▾

Create a PITermV from the given [PITerm\[\]](#) array.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PITermV(  
    PITerm[] terms  
)  
  
Public Sub New (  
    terms As PITerm()  
)  
  
public:  
PITermV(  
    array<PITerm>^ terms  
)  
  
new :  
    terms : PITerm[] -> PITermV
```

Parameters

terms ([PITerm\[\]](#))

An array of [PITerm](#)s to build the vector.

Examples

Use of Initializing an Array in CSharp

```
P1TermV v = new P1TermV(new P1Term[] {t1, t2,
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Constructor (size)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► [PITermV\(Int32\)](#)

C# ▾

Create a vector of PI Terms with size elements

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTermV(  
    int size  
)  
  
Public Sub New (  
    size As Integer  
)  
  
public:  
PlTermV(  
    int size  
)  
  
new :  
    size : int -> PlTermV
```

Parameters

size (Int32)

The amount of PI Terms in the vector

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta@SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Equality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTermV](#) ► **Equality(PlTermV, PlTermV)**

C# ▾

[Missing <summary> documentation for
"M:SbsSW.SwiPICs.PlTermV.op_Equality(SbsSW.SwiPICs.PlTermV

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator ==(  
    PlTermV termVector1,  
    PlTermV termVector2  
)  
  
Public Shared Operator = (  
    termVector1 As PlTermV,  
    termVector2 As PlTermV  
) As Boolean  
  
public:  
static bool operator ==(  
    PlTermV termVector1,  
    PlTermV termVector2  
)  
  
static let inline (=)  
    termVector1 : PlTermV *  
    termVector2 : PlTermV : bool
```

Parameters

termVector1 ([PlTermV](#))

[Missing <param name="termVector1"/> documentation for
"M:SbsSW.SwiPICs.PItermV.op_Equality(SbsSW.SwiPICs.PItermV

termVector2 ([PItermV](#))

[Missing <param name="termVector2"/> documentation for
"M:SbsSW.SwiPICs.PItermV.op_Equality(SbsSW.SwiPICs.PItermV

Return Value

[Boolean](#)

[Missing <returns> documentation for
"M:SbsSW.SwiPICs.PItermV.op_Equality(SbsSW.SwiPICs.PItermV

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Method[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► **Equals()**

C# ▾

Members

Icon	Member	Description
=	Equals(Object)	Indicates whether this instance and a specified object are equal. (Overrides ValueType.Equals(Object) .)
=	Equals(PITermV)	Compare the size and A0 of the PTermV

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (other)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► **Equals(PITermV)**

C# ▾

Compare the size and A0 of the PTermV

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool Equals(
    PTermV other
)

Public Function Equals (
    other As PTermV
) As Boolean

public:
virtual bool Equals(
    PTermV other
) sealed

abstract Equals :
    other : PTermV -> bool
override Equals :
    other : PTermV -> bool
```

Parameters

other ([PITermV](#))

The PTermV to compare

Return Value

[Boolean](#)

Return **false** if size or A0 are not equal otherwise **true**.

Remarks

// TODO compare each PIterm in PItermV not only the references in A0

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method (obj)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► **Equals(Object)**

C# ▾

Indicates whether this instance and a specified object are equal.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override bool Equals(
    Object obj
)

Public Overrides Function Equals (
    obj As Object
) As Boolean

public:
virtual bool Equals(
    Object^ obj
) override

abstract Equals :
    obj : Object -> bool
override Equals :
    obj : Object -> bool
```

Parameters

obj ([Object](#))

Another object to compare to.

Return Value

[Boolean](#)

true if *obj* and this instance are the same type and represent the same

value; otherwise, false.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **GetHashCode Method**

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► **GetHashCode()**

C# ▾

Returns the hash code for this instance.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override int GetHashCode()  
  
Public Overrides Function GetHashCode As Integer  
  
public:  
    virtual int GetHashCode() override  
  
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

[Int32](#)

A 32-bit signed integer that is the hash code for this instance.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Swi-cs-pl - A CSharp class library to connect .NET languages with SWI-Prolog

Inequality Operator

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTermV](#) ► [Inequality\(PlTermV, PlTermV\)](#)

C# ▾

[Missing <summary> documentation for
"M:SbsSW.SwiPICs.PlTermV.op_Inequality(SbsSW.SwiPICs.PlTermV, SbsSW.SwiPICs.PlTermV)"

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public static bool operator !=(
    PlTermV termVector1,
    PlTermV termVector2
)

Public Shared Operator <> (
    termVector1 As PlTermV,
    termVector2 As PlTermV
) As Boolean

public:
static bool operator !=(
    PlTermV termVector1,
    PlTermV termVector2
)

static let inline (<>)
    termVector1 : PlTermV *
    termVector2 : PlTermV : bool
```

Parameters

termVector1 ([PlTermV](#))

[Missing <param name="termVector1"/> documentation for
"M:SbsSW.SwiPICs.PI.TermV.op_Inequality(SbsSW.SwiPICs.PI"

termVector2 ([PI.TermV](#))

[Missing <param name="termVector2"/> documentation for
"M:SbsSW.SwiPICs.PI.TermV.op_Inequality(SbsSW.SwiPICs.PI"

☒ Return Value

[Boolean](#)

[Missing <returns> documentation for
"M:SbsSW.SwiPICs.PI.TermV.op_Inequality(SbsSW.SwiPICs.PI.Term

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Property (index)

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PlTermV](#) ► **Item[Int32]**

C# ▾

A zero based list

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTerm this[
    int index
] { get; set; }

Public Default Property Item (
    index As Integer
) As PlTerm
    Get
    Set

public:
property PlTerm default[int index] {
    PlTerm get (int index);
    void set (int index, PlTerm value);
}

member Item : PlTerm with get, set
```

Parameters

index ([Int32](#))

Return Value

[PlTerm](#)

The PlTerm for the given index

Exceptions

Exception	Condition
ArgumentOutOfRangeException	Is thrown if (index < 0 index >= Size)
PreconditionException	Is thrown if the operator is used on an uninitialized PIterm

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► [PITermV](#) ► **Size**

C# ▾

Get the size of a PItermV

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public int Size { get; }

Public ReadOnly Property Size As Integer
    Get

public:
property int Size {
    int get ();
}

member Size : int with get
```

Property Value

[Int32](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Enumeration

[SwiPICs interface](#) ► [SbsSW.SwiPICs](#) ► **PIType**

C# ▾

Obtain the type of a term, which should be a term returned by one of the other interface predicates or passed as an argument. The function returns the type of the Prolog term. The type identifiers are listed below.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public enum PIType

Public Enumeration PIType

public enum class PIType

type PIType
```

Members

Member	Description
PILUnknown	0 - PL_UNKNOWN: Undefined
PILVariable	1 - PL_VARIABLE: An unbound variable. The value of term as such is a unique identifier for the variable.
PILAtom	2 - PL_ATOM: A Prolog atom.
PILInteger	3 - PL_INTEGER: A Prolog integer.
PILFloat	4 - PL_FLOAT: A Prolog floating point number.
PILString	5 - PL_STRING: A Prolog string.
PITerm	6 - PL_TERM: A compound term. Note that a list is a compound term ./2.

PICodeList	14 - PL_CODE_LIST: [ascii...].
PICharList	15 - PL_CHAR_LIST: [h,e,l,l,o].

- Remarks

see [PL_term_type\(term_t\)](#) in the SWI-Prolog Manual.

- Examples

In this sample a Prolog variable is created in **PITerm t** and the [PIType](#) is checked by his integer representation and his name.

[Cop](#)

```
P1Term t = P1Term.P1Var();
Assert.AreEqual(1, (int)t.P1Type);
Assert.AreEqual(P1Type.P1Variable, t.P1Type);
```

- See Also

[PIType](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

The namespace SbsSW.SwiPICs.Callback provides the delegates to register .NET methods to be called from [SWI-Prolog](#)

■ Types

All Types	Enumerations	Delegates
Icon	Type	Description
	DelegateParameter0	Provide a predefined Delegate to register a C# method to be called from SWI-Prolog
	DelegateParameter1	Provide a predefined Delegate to register a C# method to be called from SWI-Prolog
	DelegateParameter2	Provide a predefined Delegate to register a C# method to be called from SWI-Prolog
	DelegateParameter3	Provide a predefined Delegate to register a C# method to be called from SWI-Prolog

	DelegateParameterBacktrack	NOT IMPLEMENTED YET
		<p>For details to implement see 9.6.17 Registering Foreign Predicates</p> <p>see also PL_foreign_control</p>
	DelegateParameterVarArgs	With this delegate you can build call-back predicate with a variable amount of parameters.
	PIForeignSwitches	Flags that are responsible for the foreign predicate parameters

Remarks

Note:

It is only possible to call static methods

See Also

[RegisterForeign\(String, Delegate\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

DelegateParameter0 Delegate

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Callback](#) ►

DelegateParameter0

C# ▾

Provide a predefined Delegate to register a C# method to be called from SWI-Prolog

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public delegate bool DelegateParameter0()  
  
Public Delegate Function DelegateParameter0 A  
  
public delegate bool DelegateParameter0()  
  
type DelegateParameter0 =  
    delegate of unit -> bool
```

Return Value

[Boolean](#)

true for succeeding otherwise false for fail

Examples

This example is for [DelegateParameter2](#) and shows how to call a C# method with two parameters.

For other samples see the source file CallbackForeignPredicate.cs in the TestSwiPI VS2008 test project.

```
public void t_in_out()
```

```
{  
    Delegate replaceDelegate = new DelegatePa  
    PlEngine.RegisterForeign(replaceDelegate)  
    for (int i = 1; i < 10; i++)  
    {  
        var arg = new PlTermV(new PlTerm("tes  
        PlQuery.PlCall("atom_replace", arg);  
        Assert.AreEqual("test_xx_f", arg[1].T  
    }  
}  
public static bool atom_replace(PlTerm atomIn  
{  
    return atomOut.Unify(atomIn.ToString()).Re  
}
```

See Also

[RegisterForeign\(Delegate\)](#)

[PlEngine\(\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



DelegateParameter1 Delegate

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Callback](#) ►

DelegateParameter1

C# ▾

Provide a predefined Delegate to register a C# method to be called from SWI-Prolog

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public delegate bool DelegateParameter1(
    PlTerm term
)

Public Delegate Function DelegateParameter1 (
    term As PlTerm
) As Boolean

public delegate bool DelegateParameter1(
    PlTerm term
)

type DelegateParameter1 =
    delegate of
        term : PlTerm -> bool
```

Parameters

term ([PlTerm](#))

Return Value

[Boolean](#)

true for succeeding otherwise false for fail

Examples

See also the example in [DelegateParameter2](#).

```
[TestMethod]
public void t_creating_a_list()
{
    Delegate d = new DelegateParameter1(create
PlEngine.RegisterForeign(d);
    for (int i = 1; i < 10; i++)
    {
        PlTerm t = PlQuery.PlCallQuery("creat
        Assert.AreEqual("[a,b,c]", t.ToString
    }
}
public static bool create_list(PlTerm list)
{
    return list.Unify(new PlTerm("[a,b,c]"));
}
```

See Also

[RegisterForeign\(Delegate\)](#)

[PlEngine\(\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **DelegateParameter2 Delegate**

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Callback](#) ►

DelegateParameter2

C# ▾

Provide a predefined Delegate to register a C# method to be called from SWI-Prolog

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public delegate bool DelegateParameter2(
    PlTerm term1,
    PlTerm term2
)

Public Delegate Function DelegateParameter2 (
    term1 As PlTerm,
    term2 As PlTerm
) As Boolean

public delegate bool DelegateParameter2(
    PlTerm term1,
    PlTerm term2
)

type DelegateParameter2 =
    delegate of
        term1 : PlTerm *
        term2 : PlTerm -> bool
```

Parameters

term1 (PlTerm)

term2 ([PlTerm](#))

- Return Value

[Boolean](#)

true for succeeding otherwise false for fail

- Examples

This example is for **DelegateParameter2** and shows how to call a C# method with two parameters.

For other samples see the source file `CallbackForeignPredicate.cs` in the `TestSwIPI` VS2008 test project.

```
public void t_in_out()
{
    Delegate replaceDelegate = new DelegatePa
PlEngine.RegisterForeign(replaceDelegate)
for (int i = 1; i < 10; i++)
{
    var arg = new PlTermV(new PlTerm("tes
PlQuery.PlCall("atom_replace", arg);
Assert.AreEqual("test_xx_f", arg[1].T
}
}
public static bool atom_replace(PlTerm atomIn
{
    return atomOut.Unify(atomIn.ToString()).Re
}
```

- See Also

[RegisterForeign\(Delegate\)](#)

[PlEngine\(\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



DelegateParameter3 Delegate

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Callback](#) ►

DelegateParameter3

C# ▾

Provide a predefined Delegate to register a C# method to be called from SWI-Prolog

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```

public delegate bool DelegateParameter3(
    PlTerm term1,
    PlTerm term2,
    PlTerm term3
)

Public Delegate Function DelegateParameter3 (
    term1 As PlTerm,
    term2 As PlTerm,
    term3 As PlTerm
) As Boolean

public delegate bool DelegateParameter3(
    PlTerm term1,
    PlTerm term2,
    PlTerm term3
)

type DelegateParameter3 =
    delegate of
        term1 : PlTerm *
        term2 : PlTerm *
        term3 : PlTerm -> bool

```

[-] Parameters

term1 ([PlTerm](#))

term2 ([PlTerm](#))

term3 ([PlTerm](#))

[-] Return Value

[Boolean](#)

true for succeeding otherwise false for fail

[-] Examples

This example is for [DelegateParameter2](#) and shows how to call a C# method with two parameters.

For other samples see the source file CallbackForeignPredicate.cs in the TestSwIPI VS2008 test project.

```
public void t_in_out()
{
    Delegate replaceDelegate = new DelegatePa
    PlEngine.RegisterForeign(replaceDelegate)
    for (int i = 1; i < 10; i++)
    {
        var arg = new PlTermV(new PlTerm("tes
        PlQuery.PlCall("atom_replace", arg);
        Assert.AreEqual("test_xx_f", arg[1].T
    }
}
public static bool atom_replace(PlTerm atomIn
{
    return atomOut.Unify(atomIn.ToString()).Re
```

}

■ See Also

[RegisterForeign\(Delegate\)](#)
[PIEngine\(\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



NOT IMPLEMENTED YET

For details to implement see [9.6.17 Registering Foreign Predicates](#)

see also [PL_foreign_control](#)

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public delegate int DelegateParameterBacktrack
    PlTerm term1,
    PlTerm term2,
    IntPtr control
)

Public Delegate Function DelegateParameterBacktrack(
    term1 As PlTerm,
    term2 As PlTerm,
    control As IntPtr
) As Integer

public delegate int DelegateParameterBacktrack
    PlTerm term1,
    PlTerm term2,
    IntPtr control
)

type DelegateParameterBacktrack =
    delegate of
```

```
term1 : PlTerm *
term2 : PlTerm *
control : IntPtr -> int
```

[-] Parameters

***term1* ([PlTerm](#))**

TODO

***term2* ([PlTerm](#))**

TODO

***control* ([IntPtr](#))**

TODO

[-] Return Value

[Int32](#)

TODO

[-] Examples

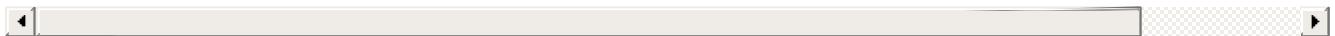
TODO

See "t_backtrack" in [TestSwiPl.CallbackForeignPredicate.cs](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **DelegateParameterVarArgs Delegate**

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Callback](#) ► [DelegateParameterVarArgs](#)

C# ▾

With this delegate you can build a call-back predicate with a variable amount of parameters.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public delegate bool DelegateParameterVarArgs
    PITermV termVector
)

Public Delegate Function DelegateParameterVar
    termVector As PItermV
) As Boolean

public delegate bool DelegateParameterVarArgs
    PItermV termVector
)

type DelegateParameterVarArgs =
    delegate of
        termVector : PItermV -> bool
```

Parameters

termVector ([PITermV](#))

The termVector representing the arguments which can be accessed by the indexer of PItermV see [PITermV](#). The amount of parameters is in [Size](#)

Return Value

Boolean

True for succeeding otherwise false for fail

■ Remarks

TODO: This do *NOT* work on 64-Bit systems. Hope to Fix this in the future.

It seems to be impossible to marshal two parameter which are bigger than 8 byte into one struct. Perhaps there is a way in CLI :-(

The problem are the parameters of the call back method. These are in (foreign_t (f)(term_t t0, int a, control_t ctx) (see SWI-cpp.h) If we provide

DelegateParameterVarArgs(P1Term term, int arity); and do in the callback Method something like

```
public static bool my_call_back(P1Term term,
{
    P1TermV args = new P1TermV(term, arity);
}
```

every thing work fine. The drawback is this ugly ctor. It might be better to do P1TermV args = P1TermV.VarArgs(term, arity); with a strong recommendation to use it *ONLY* in this call back scenario.

■ Examples

```
public void t_varargs()
{
    Delegate d = new DelegateParameterVarArgs
    PlEngine.RegisterForeign("my_concat_atom"
```

```

PlEngine.RegisterForeign("my_concat_atom"
for (int i = 1; i < 10; i++)
{
    PlTerm t = PlQuery.PlCallQuery("my_co
    Assert.AreEqual("abc", t.ToString(),
    t = PlQuery.PlCallQuery("my_concat_at
    Assert.AreEqual("abcdef", t.ToString(
}
public static bool my_concat_atom(PlTermV ter
{
    System.Diagnostics.Debug.Print("my_concat

    int arity = term1.Size;
    string sRet = "";
    PlTerm termOut = term1[arity -1];

    for (int i = 0; i < arity-1; i++)
    {
        sRet += term1[i].ToString();
    }
    termOut.Unify(sRet);
    return true;
}

```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
 (1.1.60605.0)



PIForeignSwitches Enumeration

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Callback](#) ► **PIForeignSwitches**

C# ▾

Flags that are responsible for the foreign predicate parameters

Declaration Syntax

C#

Visual Basic

Visual C++

F#

[FlagsAttribute]

public enum PIForeignSwitches

<FlagsAttribute>

Public Enumeration PIForeignSwitches

[FlagsAttribute]

public enum class PIForeignSwitches

[<FlagsAttribute>]

type PIForeignSwitches

Members

Member	Description
None	0 - PL_FA_NOTHING: no flags.
NoTrace	1 - PL_FA_NOTRACE: Predicate cannot be seen in the tracer.
Transparent	2 - PL_FA_TRANSPARENT: Predicate is module transparent.
Nondeterministic	4 - PL_FA_NONDETERMINISTIC: Predicate is non-deterministic. See also <code>PL_retry()</code> .
VarArgs	8 - PL_FA_VARARGS: (Default) Use alternative calling convention.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

These are the namespace comments for **SbsSW.SwiPICs.Exceptions**

The namespace SbsSW.SwiPICs.Exceptions provides the Exception classes to catch a prolog exception see [SWI-Prolog Manual - 4.9 ISO compliant Exception handling](#)

Prolog exceptions are mapped to C# exceptions using the subclass PIException of [Exception](#) to represent the Prolog exception term.

All type-conversion functions of the interface raise Prolog-compliant exceptions, providing decent error-handling support at no extra work for the programmer.

For some commonly used exceptions, subclasses of PIException have been created to exploit both their constructors for easy creation of these exceptions as well as selective trapping in C#.

Currently, these are [PITypeException](#) and [PIDomainException](#).

[-] Types

All Types

Classes

Icon	Type	Description
	PIDomainException	A domain exception expresses that a term satisfies the basic Prolog type expected, but is unacceptable to the restricted domain expected by some operation.
	PIException	This class is the base class to catch

		exceptions thrown by prolog in C#.
	PILibException	This exception is thrown if something in the interface went wrong.
	PITypeException	A type error expresses that a term does not satisfy the expected basic Prolog type.

■ Remarks

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Prolog **PIDomainException Class**

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ►

PIDomainException

C# ▾

A domain exception expresses that a term satisfies the basic Prolog type expected, but is unacceptable to the restricted domain expected by some operation.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
[SerializableAttribute]
public class PIDomainException : PLEnvironmentException

<SerializableAttribute>
Public Class PIDomainException
    Inherits PLEnvironmentException

[SerializableAttribute]
public ref class PIDomainException : public PLEnvironmentException

[<SerializableAttribute>]
type PIDomainException =
    class
        inherit PLEnvironmentException
    end
```

Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declared

Inherited

Icon Member

Description

	PIDomainException()	This class is the base class to catch exceptions thrown by prolog in C#.
	PIDomainException(String)	This class is the base class to catch exceptions thrown by prolog in C#.
	PIDomainException(String, Exception)	This class is the base class to catch exceptions thrown by prolog in C#.
	PIDomainException(SerializationInfo, StreamingContext)	This class is the base class to catch exceptions thrown by prolog in C#.
	PIDomainException(PITerm)	This class is the base class to catch exceptions thrown by prolog in C#.
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception.

		(Inherited from Exception .)
≡	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
≡	Finalize()	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
≡	GetBaseException()	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception .)
≡	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object .)
≡	GetObjectData(SerializationInfo, StreamingContext)	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from PIException .)
≡	GetType()	Gets the runtime type of the current instance.

		(Inherited from Exception .)
	HelpLink	Gets or sets a link to the help file associated with this exception.
		(Inherited from Exception .)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception.
		(Inherited from Exception .)
	InnerException	Gets the Exception instance that caused the current exception.
		(Inherited from Exception .)
	MemberwiseClone()	Creates a shallow copy of the current Object .
		(Inherited from Object .)
	Message	Gets a message that describes the current exception.
		(Inherited from PIException .)
	MessagePI	provide sometimes some additional information about the exceptions reason.
		(Inherited from PIException .)
	PIThrow()	Generate an exception (as

		throw/1) and return false .
		(Inherited from PIException .)
	Source	Gets or sets the name of the application or the object that causes the error.
		(Inherited from Exception .)
	StackTrace	Gets a string representation of the frames on the call stack at the time the current exception was thrown.
		(Inherited from Exception .)
	TargetSite	Gets the method that throws the current exception.
		(Inherited from Exception .)
	Term	Get the PITerm of this exception.
		(Inherited from PIException .)
	Throw()	Throw this PIException.
		(Inherited from PIException .)
	ToString()	The exception is translated into a message as produced by <code>print_message/2</code> . The character data is stored in

ring.

(Inherited from
[PIException](#).)

■ Examples

For example, the standard Prolog open/3 call expect an IO-Mode (read, write, append, ...). If an integer is provided, this is a type error, if an ator other than one of the defined IO-modes is provided it is a domain error.

```
public void prolog_domain_exception_sample()
{
    try
    {
        Assert.IsTrue(PlQuery.PlCall("open(te
    }
    catch (PlDomainException ex)
    {
        Assert.AreEqual("open/3: Domain error
    }
}
```

■ Inheritance Hierarchy

[Object](#)
└ [Exception](#)
 └ [PIException](#)
 └ **PlDomainException**

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



PIDomainException Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ►

[PIDomainException](#) ► [PIDomainException\(\)](#)

C# ▾

Members

Icon	Member	Description
≡	PIDomainException()	This class is the base class to catch exceptions thrown by prolog in C#.
≡	PIDomainException(String)	This class is the base class to catch exceptions thrown by prolog in C#.
≡	PIDomainException(String, Exception)	This class is the base class to catch exceptions thrown by prolog in C#.
💡	PIDomainException(SerializationInfo, StreamingContext)	This class is the base class to catch exceptions thrown by prolog in C#.
≡	PIDomainException(PITerm)	This class is the base class to catch exceptions thrown by prolog in C#.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PIDomainException Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ►

[PIDomainException](#) ► [PIDomainException\(\)](#)

C# ▾

This class is the base class to catch exceptions thrown by prolog in C#.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlDomainException()

Public Sub New

public:
PlDomainException()

new : unit -> PlDomainException
```

Examples

```
public void prolog_exception_sample()
{
    const string exceptionText = "test_except"
    Assert.IsTrue(PlQuery.PlCall("assert( (te
try
{
    Assert.IsTrue(PlQuery.PlCall("test_th
}
catch (PlException ex)
{
    Assert.AreEqual(exceptionText, ex.Ter
```

```
        Assert.AreEqual("Unknown message: " +  
    }  
}
```

See Also

[PITypeException](#)

[SWI-Prolog Manual - 4.9 ISO compliant Exception handling](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PIDomainException** Constructor (term)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ►

[PIDomainException](#) ► **PIDomainException(PITerm)**

C# ▾

This class is the base class to catch exceptions thrown by prolog in C#.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIDomainException(
    PITerm term
)

Public Sub New (
    term As PITerm
)

public:
PIDomainException(
    PITerm term
)

new :
    term : PITerm -> PIDomainException
```

Parameters

term ([PITerm](#))

[Missing <param name="term"/> documentation for
 "M:SbsSW.SwiPICs.Exceptions.PIDomainException.#ctor(Sbs:

Examples

```
public void prolog_exception_sample()
{
    const string exceptionText = "test_except"
    Assert.IsTrue(PlQuery.PlCall("assert( (te
        try
    {
        Assert.IsTrue(PlQuery.PlCall("test_th
    }
    catch (PlException ex)
    {
        Assert.AreEqual(exceptionText, ex.Ter
        Assert.AreEqual("Unknown message: " +
    }
}
}
```

See Also

[PITypeException](#)

[SWI-Prolog Manual - 4.9 ISO compliant Exception handling](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PIDomainException** Constructor (info, context)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ►
[PIDomainException](#) ► **PIDomainException(SerializationInfo, StreamingContext)**

C# ▾

This class is the base class to catch exceptions thrown by prolog in C#.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
protected PIDomainException(
    SerializationInfo info,
    StreamingContext context
)

Protected Sub New (
    info As SerializationInfo,
    context As StreamingContext
)

protected:
PIDomainException(
    SerializationInfo^ info,
    StreamingContext context
)

new :
    info : SerializationInfo *
    context : StreamingContext -> PIDomai
```

Parameters

[*info* \(*SerializationInfo*\)](#)

[Missing <param name="info"/> documentation for
"M:SbsSW.SwiPICs.Exceptions.PIDomainException.#ctor(Sys

[*context* \(*StreamingContext*\)](#)

[Missing <param name="context"/> documentation for
"M:SbsSW.SwiPICs.Exceptions.PIDomainException.#ctor(Sys

[**Examples**](#)

```
public void prolog_exception_sample()
{
    const string exceptionText = "test_except"
    Assert.IsTrue(PlQuery.PlCall("assert( (te
        try
    {
        Assert.IsTrue(PlQuery.PlCall("test_th
    }
    catch (PlException ex)
    {
        Assert.AreEqual(exceptionText, ex.Ter
        Assert.AreEqual("Unknown message: " +
    }
}
```

[**See Also**](#)

[PITypeException](#)

[SWI-Prolog Manual - 4.9 ISO compliant Exception handling](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PIDomainException** Constructor (message)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ►

[PIDomainException](#) ► **PIDomainException(String)**

C# ▾

This class is the base class to catch exceptions thrown by prolog in C#.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIDomainException(
    string message
)

Public Sub New (
    message As String
)

public:
PIDomainException(
    String^ message
)

new :
    message : string -> PIDomainException
```

Parameters

message ([String](#))

[Missing <param name="message"/> documentation for
 "M:SbsSW.SwiPICs.Exceptions.PIDomainException.#ctor(Sys1

Examples

```
public void prolog_exception_sample()
{
    const string exceptionText = "test_except"
    Assert.IsTrue(PlQuery.PlCall("assert( (te
        try
    {
        Assert.IsTrue(PlQuery.PlCall("test_th
    }
    catch (PlException ex)
    {
        Assert.AreEqual(exceptionText, ex.Ter
        Assert.AreEqual("Unknown message: " +
    }
}
}
```

See Also

[PlTypeException](#)

[SWI-Prolog Manual - 4.9 ISO compliant Exception handling](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PIDomainException** Constructor (message, innerException)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ►
[PIDomainException](#) ► **PIDomainException(String, Exception)**

C# ▾

This class is the base class to catch exceptions thrown by prolog in C#.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIDomainException(
    string message,
    Exception innerException
)

Public Sub New (
    message As String,
    innerException As Exception
)

public:
PIDomainException(
    String^ message,
    Exception^ innerException
)

new :
    message : string *
    innerException : Exception -> PIDomai
```

Parameters

message (String)

[Missing <param name="message"/> documentation for
"M:SbsSW.SwiPICs.Exceptions.PIDomainException.#ctor(Sys

innerException ([Exception](#))

[Missing <param name="innerException"/> documentation for
"M:SbsSW.SwiPICs.Exceptions.PIDomainException.#ctor(Sys

□ Examples

```
public void prolog_exception_sample()
{
    const string exceptionText = "test_except
Assert.IsTrue(PlQuery.PlCall("assert( (te
try
{
    Assert.IsTrue(PlQuery.PlCall("test_th
}
catch (PlException ex)
{
    Assert.AreEqual(exceptionText, ex.Ter
    Assert.AreEqual("Unknown message: " +
}
}
```

□ See Also

[PITypeException](#)

[SWI-Prolog Manual - 4.9 ISO compliant Exception handling](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0

(1.1.60605.0)



PIException Class

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► **PIException**

C# ▾

This class is the base class to catch exceptions thrown by prolog in C#.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
[SerializableAttribute]
public class PIException : Exception

<SerializableAttribute>
Public Class PIException
    Inherits Exception

[SerializableAttribute]
public ref class PIException : public Exception

[<SerializableAttribute>]
type PIException =
    class
        inherit Exception
    end
```

Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declares

Inherited

Icon

Member

Description



[PIException\(\)](#)

Initializes a new instance of the

		class.
≡	PIException(String)	Initializes a new instance of the class with a specified error message.
≡	PIException(String, Exception)	Initializes a new instance of the class with a specified error message and a reference to the inner exception that caused this exception.
💡	PIException(SerializationInfo, StreamingContext)	Initializes a new instance of the class with serialized data.
≡	PIException(PITerm)	To catch a exception thrown by For a example see PIException .
📁	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception .)
≡	Equals(Object)	Determines whether the specified object is equal to the current Object . (Inherited from Object .)
💡	Finalize()	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.

		collection.
		(Inherited from Object .)
	GetBaseException()	When overridden in a derived class, returns the Exception that is the base of one or more subsequent exceptions.
		(Inherited from Exception .)
	GetHashCode()	Serves as a hash function for a type.
		(Inherited from Object .)
	GetObjectData(SerializationInfo, StreamingContext)	When overridden in a derived class, sets the SerializationInfo with information about the exception.
		(Overrides Exception.GetObjectData(SerializationInfo, StreamingContext) .)
	GetType()	Gets the runtime type of the current instance.
		(Inherited from Exception .)
	HelpLink	Gets or sets a link to the help file associated with this exception.
		(Inherited from Exception .)
	HResult	Gets or sets HRESULT, a code-based value that is assigned to a specific exception.
		(Inherited from Exception .)
	InnerException	Gets the Exception instance that is the current exception.

		(Inherited from Exception .)
	MemberwiseClone()	Creates a shallow copy of the current Object .
		(Inherited from Object .)
	Message	Gets a message that describes the exception.
		(Overrides Exception.Message .)
	MessagePI	Provide sometimes some additional information about the exception.
	PIThrow()	Generate an exception (as throwing <code>false</code> return <code>false</code>).
	Source	Gets or sets the name of the application or the object that causes the error.
		(Inherited from Exception .)
	StackTrace	Gets a string representation of the stack on the call stack at the time the exception was thrown.
		(Inherited from Exception .)
	TargetSite	Gets the method that throws the exception.
		(Inherited from Exception .)
	Term	Get the PITerm of this exception.
	Throw()	Throw this PIException .

 ToString()	<p>The exception is translated into as produced by print_message, character data is stored in a ring (Overrides Exception.ToString())</p>
--	---

Examples

```
public void prolog_exception_sample()
{
    const string exceptionText = "test_except
Assert.IsTrue(PlQuery.PlCall("assert( (te
try
{
    Assert.IsTrue(PlQuery.PlCall("test_th
}
catch (PlException ex)
{
    Assert.AreEqual(exceptionText, ex.Ter
    Assert.AreEqual("Unknown message: " +
}
}
```

Inheritance Hierarchy

- [Object](#)
- └ [Exception](#)
 - └ [PIException](#)
 - ├ [PIDomainException](#)
 - └ [PITypeException](#)

See Also

[PITypeException](#)

[SWI-Prolog Manual - 4.9 ISO compliant Exception handling](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PIException** Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► [PIException\(\)](#)

C# ▾

Members

Icon	Member	Description
≡	PIException()	Initializes a new instance of the Exception class.
≡	PIException(String)	Initializes a new instance of the Exception class with a specified error message.
≡	PIException(String, Exception)	Initializes a new instance of the Exception class with a specified error message and a reference to the inner exception that is the cause of this exception.
💡	PIException(SerializationInfo, StreamingContext)	Initializes a new instance of the Exception class with serialized data.
≡	PIException(PITerm)	To catch a exception thrown by prolog For a example see PIException .

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PIException Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► [PIException\(\)](#)

C# ▾

Initializes a new instance of the [Exception](#) class.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIException()  
  
Public Sub New  
  
public:  
PIException()  
  
new : unit -> PIException
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PIException Constructor (term)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► [PIException\(PITerm\)](#)

C# ▾

To catch a exception thrown by prolog

For a example see [PIException](#).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIException(  
    PITerm term  
)  
  
Public Sub New (  
    term As PITerm  
)  
  
public:  
PIException(  
    PITerm term  
)  
  
new :  
    term : PITerm -> PIException
```

Parameters

term (PITerm)

A PITerm containing the Prolog exception

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PIException Constructor (info, context)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► [PIException\(SerializationInfo, StreamingContext\)](#)

C# ▾

Initializes a new instance of the [Exception](#) class with serialized data.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```

protected PIException(
    SerializationInfo info,
    StreamingContext context
)

Protected Sub New (
    info As SerializationInfo,
    context As StreamingContext
)

protected:
PIException(
    SerializationInfo^ info,
    StreamingContext context
)

new :
    info : SerializationInfo *
    context : StreamingContext -> PIExcep

```

Parameters

info ([SerializationInfo](#))

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

context ([StreamingContext](#))

The [StreamingContext](#) that contains contextual information about the source or destination.

Exceptions

Exception	Condition
ArgumentNullException	The <i>info</i> parameter is null.
SerializationException	The class name is null or HResult is zero (0).

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



C# ▾

Initializes a new instance of the [Exception](#) class with a specified error message.

[-] Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIException(  
    string message  
)  
  
Public Sub New (  
    message As String  
)  
  
public:  
PIException(  
    String^ message  
)  
  
new :  
    message : string -> PIException
```

[-] Parameters

message ([String](#))

The message that describes the error.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0

(1.1.60605.0)

PIException Constructor (message, innerException)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► [PIException\(String, Exception\)](#)

C# ▾

Initializes a new instance of the [Exception](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIException(
    string message,
    Exception innerException
)

Public Sub New (
    message As String,
    innerException As Exception
)

public:
PIException(
    String^ message,
    Exception^ innerException
)

new :
    message : string *
    innerException : Exception -> PIExcep
```

Parameters

***message* ([String](#))**

The error message that explains the reason for the exception.

***innerException* ([Exception](#))**

The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog
GetObjectData Method (info, context)[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ►**GetObjectData(SerializationInfo, StreamingContext)**

C# ▾

When overridden in a derived class, sets the [SerializationInfo](#) with information about the exception.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)

Public Overrides Sub GetObjectData (
    info As SerializationInfo,
    context As StreamingContext
)

public:
virtual void GetObjectData(
    SerializationInfo^ info,
    StreamingContext context
) override

abstract GetObjectData :
    info : SerializationInfo *
    context : StreamingContext -> unit
override GetObjectData :
    info : SerializationInfo *
    context : StreamingContext -> unit
```

[-] Parameters

info (SerializationInfo)

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

context (StreamingContext)

The [StreamingContext](#) that contains contextual information about the source or destination.

[-] Exceptions

Exception	Condition
ArgumentNullException	The <i>info</i> parameter is a null reference (Nothing in Visual Basic).
SecurityException	The caller does not have the required permission.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► **Message**

C# ▾

Gets a message that describes the current exception.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override string Message { get; }
```

```
Public Overrides ReadOnly Property Message As
    Get
```

```
public:
virtual property String^ Message {
    String^ get () override;
}
```

```
abstract Message : string with get
override Message : string with get
```

Return Value

[String](#)

The error message that explains the reason for the exception, or an empty string("")).

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)





Prolog **MessagePl Property**

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► **MessagePl**

C# ▾

provide sometimes some additional information about the exceptions reason.

[-] Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public string MessagePl { get; }
```

```
Public ReadOnly Property MessagePl As String
    Get
```

```
public:
property String^ MessagePl {
    String^ get ();
}
```

```
member MessagePl : string with get
```

[-] Property Value

[String](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► [PIThrow\(\)](#)

C# ▾

Generate an exception (as throw/1) and return **false**.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public bool PIThrow()  
  
Public Function PIThrow As Boolean  
  
public:  
bool PIThrow()  
  
member PIThrow : unit -> bool
```

Return Value

[Boolean](#)

Generate an exception (as throw/1) and return **false**.

Remarks

Used in the PREDICATE() wrapper to pass the exception to Prolog. See [PL_raise_exception\(\)](#).

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Property

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► [Term](#)

C# ▾

Get the [PITerm](#) of this exception.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PIterm Term { get; }

Public ReadOnly Property Term As PIterm
    Get

public:
property PIterm Term {
    PIterm get ();
}

member Term : PIterm with get
```

Property Value

[PITerm](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► **Throw()**

C# ▾

Throw this PIException.

[-] Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public void Throw()

Public Sub Throw

public:
void Throw()

member Throw : unit -> unit
```

[-] Remarks

see [http://www.swi-prolog.org/packages/pl2cpp.html#cppThrow\(\)](http://www.swi-prolog.org/packages/pl2cpp.html#cppThrow())

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Method

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PIException](#) ► [ToString\(\)](#)

C# ▾

The exception is translated into a message as produced by print_message/2. The character data is stored in a ring.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override string ToString()  
  
Public Overrides Function ToString As String  
  
public:  
    virtual String^ ToString() override  
  
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

[String](#)

A textual description of the Exception

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PILibException Class**

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► **PILibException**

C# ▾

This exception is thrown if something in the interface went wrong.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
[SerializableAttribute]
public class PILibException : Exception

<SerializableAttribute>
Public Class PILibException
    Inherits Exception

[SerializableAttribute]
public ref class PILibException : public Exce

[<SerializableAttribute>]
type PILibException =
    class
        inherit Exception
    end
```

Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static



Declared



Inherited

Icon

Icon	Member	Description
	PILibException()	Initializes a new instance of the class.

	PILibException(String)	Initializes a new instance of the class with a specified error message.
	PILibException(String, Exception)	Initializes a new instance of the class with a specified error message and a reference to the inner exception that is the cause of this exception.
	PILibException(SerializationInfo, StreamingContext)	Initializes a new instance of the class with serialized data.
	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception .)
	Equals(Object)	Determines whether the specified object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
	GetBaseException()	When overridden in a derived class, returns the Exception that is the cause of one or more subsequent exceptions. (Inherited from Exception .)

	GetHashCode()	Serves as a hash function for a type. (Inherited from Object .)
	GetObjectData(SerializationInfo, StreamingContext)	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Overrides Exception.GetObjectData(SerializationInfo, StreamingContext) .)
	GetType()	Gets the runtime type of the current instance. (Inherited from Exception .)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception .)
	HResult	Gets or sets HRESULT, a code-based value that is assigned to a specific exception. (Inherited from Exception .)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception .)
	MemberwiseClone()	Creates a shallow copy of the current object. (Inherited from Object .)
	Message	Gets a message that describes the exception.

		(Inherited from Exception .)
	Source	Gets or sets the name of the application or the object that causes the error.
		(Inherited from Exception .)
	StackTrace	Gets a string representation of the exception on the call stack at the time the exception was thrown.
		(Inherited from Exception .)
	TargetSite	Gets the method that throws the exception.
		(Inherited from Exception .)
	ToString()	Creates and returns a string representation of the current exception.
		(Inherited from Exception .)

Inheritance Hierarchy

[Object](#)

└ [Exception](#)

 └ **PILibException**

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PILibException** Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PILibException](#) ► [PILibException\(\)](#)

C# ▾

Members

Icon	Member	Description
≡	PILibException()	Initializes a new instance of the Exception class.
≡	PILibException(String)	Initializes a new instance of the Exception class with a specified error message.
≡	PILibException(String, Exception)	Initializes a new instance of the Exception class with a specified error message and a reference to the inner exception that is the cause of this exception.
💡	PILibException(SerializationInfo, StreamingContext)	Initializes a new instance of the Exception class with serialized data.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PILibException** Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PILibException](#) ► [PILibException\(\)](#)

C# ▾

Initializes a new instance of the [Exception](#) class.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PILibException()  
  
Public Sub New  
  
public:  
PILibException()  
  
new : unit -> PILibException
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PILibException** Constructor (info, context)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PILibException](#) ► [PILibException\(SerializationInfo, StreamingContext\)](#)

C#

Initializes a new instance of the [Exception](#) class with serialized data.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```

protected PILibException(
    SerializationInfo info,
    StreamingContext context
)

Protected Sub New (
    info As SerializationInfo,
    context As StreamingContext
)

protected:
PILibException(
    SerializationInfo^ info,
    StreamingContext context
)

new :
    info : SerializationInfo *
    context : StreamingContext -> PILibEx

```

Parameters

info ([SerializationInfo](#))

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

context ([StreamingContext](#))

The [StreamingContext](#) that contains contextual information about the source or destination.

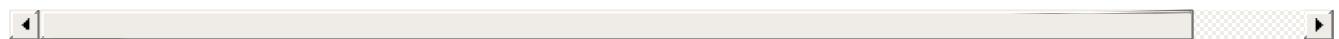
Exceptions

Exception	Condition
ArgumentNullException	The <i>info</i> parameter is null.
SerializationException	The class name is null or HResult is zero (0).

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog **PILibException Constructor (message)**

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PILibException](#) ► [PILibException\(String\)](#)

C# ▾

Initializes a new instance of the [Exception](#) class with a specified error message.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PILibException(  
                      string message  
)  
  
Public Sub New (  
                 message As String  
)  
  
public:  
PILibException(  
               String^ message  
)  
  
new :  
      message : string -> PILibException
```

Parameters

message ([String](#))

The message that describes the error.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0

(1.1.60605.0)

Prolog **PILibException** Constructor (message, innerException)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PILibException](#) ► [PILibException\(String, Exception\)](#)

C# ▾

Initializes a new instance of the [Exception](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PILibException(
    string message,
    Exception innerException
)

Public Sub New (
    message As String,
    innerException As Exception
)

public:
PILibException(
    String^ message,
    Exception^ innerException
)

new :
    message : string *
    innerException : Exception -> PILibEx
```

Parameters

***message* ([String](#))**

The error message that explains the reason for the exception.

***innerException* ([Exception](#))**

The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog
GetObjectData Method (info, context)[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PILibException](#) ►**GetObjectData(SerializationInfo, StreamingContext)**

C# ▾

When overridden in a derived class, sets the [SerializationInfo](#) with information about the exception.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public override void GetObjectData(
    SerializationInfo info,
    StreamingContext context
)

Public Overrides Sub GetObjectData (
    info As SerializationInfo,
    context As StreamingContext
)

public:
virtual void GetObjectData(
    SerializationInfo^ info,
    StreamingContext context
) override

abstract GetObjectData :
    info : SerializationInfo *
    context : StreamingContext -> unit
override GetObjectData :
    info : SerializationInfo *
    context : StreamingContext -> unit
```

[-] Parameters

info (SerializationInfo)

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

context (StreamingContext)

The [StreamingContext](#) that contains contextual information about the source or destination.

[-] Exceptions

Exception	Condition
ArgumentNullException	The <i>info</i> parameter is a null reference (Nothing in Visual Basic).
SecurityException	The caller does not have the required permission.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PITypeException Class

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► **PITypeException**

C# ▾

A type error expresses that a term does not satisfy the expected basic Prolog type.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
[SerializableAttribute]
public class PITypeException : PIException

<SerializableAttribute>
Public Class PITypeException
    Inherits PIException

[SerializableAttribute]
public ref class PITypeException : public PIException

[<SerializableAttribute>]
type PITypeException =
    class
        inherit PIException
    end
```

Members

All Members

Constructors

Methods

Properties

Public

Protected

Instance

Static

Declared

Inherited

Icon

Member

Description



[PITypeException\(\)](#)

Initializes a new instance of

		the Exception class.
≡	PITypeException(String)	Initializes a new instance of the Exception class with a specified error message.
≡	PITypeException(String, Exception)	Initializes a new instance of the Exception class with a specified error message and reference to the inner exception that is the cause of this exception.
≡	PITypeException(SerializationInfo, StreamingContext)	Initializes a new instance of the Exception class with serialized data.
≡	PITypeException(PITerm)	<p>To catch a exception thrown by prolog</p> <p>For a example see PIException.</p>
≡	PITypeException(String, PITerm)	Creates an ISO standard Prolog error term expressing the expected type and actual term that does not satisfy this type.

	Data	Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception .)
	Equals(Object)	Determines whether the specified Object is equal to the current Object . (Inherited from Object .)
	Finalize()	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .)
	GetBaseException()	When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception .)
	GetHashCode()	Serves as a hash function for a particular type. (Inherited from Object .)
	GetObjectData(SerializationInfo, StreamingContext)	When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from PIException .)

	GetType()	Gets the runtime type of the current instance. (Inherited from Exception .)
	HelpLink	Gets or sets a link to the help file associated with this exception. (Inherited from Exception .)
	HResult	Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from Exception .)
	InnerException	Gets the Exception instance that caused the current exception. (Inherited from Exception .)
	MemberwiseClone()	Creates a shallow copy of the current Object . (Inherited from Object .)
	Message	Gets a message that describes the current exception. (Inherited from PIException .)
	MessagePI	provide sometimes some additional information about the exceptions reason. (Inherited from PIException .)

	PIThrow()	Generate an exception (as throw/1) and return false . (Inherited from PIException .)
	Source	Gets or sets the name of the application or the object that causes the error. (Inherited from Exception .)
	StackTrace	Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception .)
	TargetSite	Gets the method that throws the current exception. (Inherited from Exception .)
	Term	Get the PITerm of this exception. (Inherited from PIException .)
	Throw()	Throw this PIException. (Inherited from PIException .)
	ToString()	The exception is translated into a message as produced by <code>print_message/2</code> . The character data is stored in a ring. (Inherited from PIException .)

- Examples

This sample demonstrate how to catch a PITypeException in C# that is thrown somewhere int the prolog code.

```
public void prolog_type_exception_sample()
{
    try
    {
        Assert.IsTrue(PlQuery.PlCall("sumlist")
    }
    catch (PITypeException ex)
    {
        Assert.AreEqual("is/2: Arithmetic: `e
    }
}
```

[-] Inheritance Hierarchy

- [Object](#)
 - [Exception](#)
 - [PIException](#)
 - [PITypeException](#)

[-] See Also

[PITypeException](#)

[SWI-Prolog Manual - 4.9 ISO compliant Exception handling](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



PITypeException Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PITypeException](#)
 ► [PITypeException\(\)](#)

C# ▾

Members

Icon	Member	Description
≡	PITypeException()	Initializes a new instance of the Exception class.
≡	PITypeException(String)	Initializes a new instance of the Exception class with a specified error message.
≡	PITypeException(String, Exception)	Initializes a new instance of the Exception class with a specified error message and reference to the inner exception that is the cause of this exception.
💡	PITypeException(SerializationInfo, StreamingContext)	Initializes a new instance of the Exception class with serialized data.
💡	PITypeException(PITerm)	To catch a exception thrown by prolog For a example see PIException .

	<u>PITypeException(String, PITerm)</u>	Creates an ISO standard Prolog error term expressing the expected type and actual term that does not satisfy this type.
---	--	---

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to Lesta at SBS-Softwaresysteme.de

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

PITypeException Constructor

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PITypeException](#)
► [PITypeException\(\)](#)

C# ▾

Initializes a new instance of the [Exception](#) class.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PITypeException()  
  
Public Sub New  
  
public:  
PITypeException()  
  
new : unit -> PITypeException
```

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PITypeException** Constructor (term)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PITypeException](#)
► [PITypeException\(PITerm\)](#)

C# ▾

To catch a exception thrown by prolog

For a example see [PIException](#).

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PITypeException(  
    PITerm term  
)  
  
Public Sub New (  
    term As PITerm  
)  
  
public:  
PITypeException(  
    PITerm term  
)  
  
new :  
    term : PITerm -> PITypeException
```

Parameters

term ([PITerm](#))

A PITerm containing the Prolog exception

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PITypeException** Constructor (info, context)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PITypeException](#)

► **PITypeException(SerializationInfo, StreamingContext)**

C# ▾

Initializes a new instance of the [Exception](#) class with serialized data.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
protected PITypeException(
    SerializationInfo info,
    StreamingContext context
)

Protected Sub New (
    info As SerializationInfo,
    context As StreamingContext
)

protected:
PITypeException(
    SerializationInfo^ info,
    StreamingContext context
)

new :
    info : SerializationInfo *
    context : StreamingContext -> PITypeE
```

Parameters

info ([SerializationInfo](#))

The [SerializationInfo](#) that holds the serialized object data about the exception being thrown.

context ([StreamingContext](#))

The [StreamingContext](#) that contains contextual information about the source or destination.

Exceptions

Exception	Condition
ArgumentNullException	The <i>info</i> parameter is null.
SerializationException	The class name is null or HResult is zero (0).

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



PITypeException Constructor (message)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PITypeException](#)
► [PITypeException\(String\)](#)

C# ▾

Initializes a new instance of the [Exception](#) class with a specified error message.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PITypeException(  
                      string message  
)  
  
Public Sub New (  
                message As String  
)  
  
public:  
PITypeException(  
                String^ message  
)  
  
new :  
      message : string -> PITypeException
```

Parameters

message ([String](#))

The message that describes the error.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0

(1.1.60605.0)

Prolog **PlTypeException** Constructor (expected, actual)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PlTypeException](#)
 ► **PlTypeException(String, PlTerm)**

C# ▾

Creates an ISO standard Prolog error term expressing the expected type and actual term that does not satisfy this type.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PlTypeException(
    string expected,
    PlTerm actual
)

Public Sub New (
    expected As String,
    actual As PlTerm
)

public:
PlTypeException(
    String^ expected,
    PlTerm actual
)

new :
    expected : string *
    actual : PlTerm -> PlTypeException
```

Parameters

***expected* ([String](#))**

The type which was expected

actual (PITerm)

The actual term

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](http://Lesta.at/SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

Prolog **PITypeException** Constructor (message, innerException)

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Exceptions](#) ► [PITypeException](#)
 ► **PITypeException(String, Exception)**

C# ▾

Initializes a new instance of the [Exception](#) class with a specified error message and a reference to the inner exception that is the cause of this exception.

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public PITypeException(
    string message,
    Exception innerException
)

Public Sub New (
    message As String,
    innerException As Exception
)

public:
PITypeException(
    String^ message,
    Exception^ innerException
)

new :
    message : string *
    innerException : Exception -> PITypeE
```

Parameters

***message* ([String](#))**

The error message that explains the reason for the exception.

***innerException* ([Exception](#))**

The exception that is the cause of the current exception, or a null reference (Nothing in Visual Basic) if no inner exception is specified

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



The namespace SbsSW.SwiPICs.Streams provides the delegates to redirect the read and write functions of the [SWI-Prolog](#) IO Streams.

When [Initialize\(String\[\]\)](#) is called the *Sinput->functions.read is replaced by the .NET method 'Sread_function' and *Sinput->functions.write by 'Swrite_funktion'.

For further examples see the methods

[SetStreamFunctionWrite\(PIStreamType, DelegateStreamWriteFunction\)](#) and [SetStreamFunctionRead\(PIStreamType, DelegateStreamReadFunction\)](#)

[-] Types

All Types	Enumerations	Delegates
Icon	Type	Description
	DelegateStreamReadFunction	See SetStreamFunctionRead(PIStreamType, DelegateStreamReadFunction)
	DelegateStreamWriteFunction	See SetStreamFunctionWrite(PIStreamType, DelegateStreamWriteFunction)
	PIStreamType	The standard SWI-Prolog stream types (standard, input, output, error)

[-] Remarks

Note:

The reason for this is debugging.

[-] Examples

```
static internal long Swrite_function(IntPtr h
{
    string s = buf.Substring(0, (int)bufsize)
    Console.WriteLine(s);
    System.Diagnostics.Trace.WriteLine(s);
    return bufsize;
}

static internal long Sread_function(IntPtr ha
{
    throw new P1LibException("SwiPLCs: Prolog
}
```

[-] See Also

[SetStreamFunctionRead\(P1StreamType, DelegateStreamReadFunction](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)



Prolog **DelegateStreamReadFunction Delegate**

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Streams](#) ►

DelegateStreamReadFunction

C# ▾

See [SetStreamFunctionRead\(PIStreamType, DelegateStreamReadFunction\)](#)

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public delegate long DelegateStreamReadFunction(
    IntPtr handle,
    IntPtr buffer,
    long bufferSize
)

Public Delegate Function DelegateStreamReadFunction(
    handle As IntPtr,
    buffer As IntPtr,
    bufferSize As Long
) As Long

public delegate long long DelegateStreamReadFunction(
    IntPtr handle,
    IntPtr buffer,
    long long bufferSize
)

type DelegateStreamReadFunction =
    delegate of
        handle : IntPtr *
        buffer : IntPtr *
        bufferSize : int64 -> int64
```

▀ Parameters

handle ([IntPtr](#))

A C stream handle. simple ignore it.

buffer ([IntPtr](#))

A pointer to a string buffer

bufferSize ([Int64](#))

The size of the string buffer

▀ Return Value

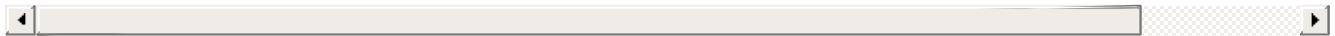
[Int64](#)

A [Delegate](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



PrologDelegateStreamWriterFunction Delegate[SwiPICs interface](#) ► [SbsSW.SwiPICs.Streams](#) ►**DelegateStreamWriterFunction**

C# ▾

See [SetStreamFunctionWrite\(PIStreamType, DelegateStreamWriterFunction\)](#)

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public delegate long DelegateStreamWriterFunct
    IntPtr handle,
    string buffer,
    long bufferSize
)

Public Delegate Function DelegateStreamWriterF
    handle As IntPtr,
    buffer As String,
    bufferSize As Long
) As Long

public delegate long long DelegateStreamWriter
    IntPtr handle,
    String^ buffer,
    long long bufferSize
)

type DelegateStreamWriterFunction =
    delegate of
        handle : IntPtr *
        buffer : string *
        bufferSize : int64 -> int64
```

▀ Parameters

handle ([IntPtr](#))

A C stream handle. simple ignore it.

buffer ([String](#))

A pointer to a string buffer

bufferSize ([Int64](#))

The size of the string buffer

▀ Return Value

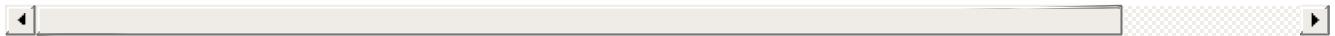
[Int64](#)

A [Delegate](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](#)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)



Prolog PIStreamType Enumeration

[SwiPICs interface](#) ► [SbsSW.SwiPICs.Streams](#) ► [PIStreamType](#)

C# ▾

The standard SWI-Prolog streams (input output error)

Declaration Syntax

C#

Visual Basic

Visual C++

F#

```
public enum PIStreamType
Public Enumeration PIStreamType
public enum class PIStreamType
type PIStreamType
```

Members

Member	Description
Input	0 - The standard input stream.
Output	1 - The standard output stream.
Error	1 - The standard error stream.

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)

Assembly: SwiPICs (Module: SwiPICs.dll) Version: 1.1.60605.0
(1.1.60605.0)

interface

SwiPICs interface

C# ▾

ist in der quelle (class NamespaceDoc) vom namespace SbsSW.SwiPICs

■ Namespaces

Namespace	Description
SbsSW.SwiPICs	<p>The online documentation home is here</p> <p>This namespace SbsSW.SwiPICs provides an .NET interface to SWI-Prolog</p>
Overview	
<p>Prolog variables are dynamically typed and all information is passed around using the C-interface type term_t which is an int. In C#, term_t is embedded in the lightweight struct PITerm. Constructors and operator definitions provide flexible operations and integration with important C#-types (string, int and double).</p> <p>The list below summarises the important classes / struct defined in the C# interface.</p>	
class / struct	Short description
PIEngine	A static class represents the prolog engine.

	<p>PITerm A struct representing prolog data.</p> <p>PITermV A vector of PITerm.</p> <p>PIQuery A class to query Prolog.</p>
SbsSW.SwiPICs.Callback	<p>The namespace SbsSW.SwiPICs.Callback provides the delegates to register .NET methods to be called from SWI-Prolog</p>
SbsSW.SwiPICs.Exceptions	<p>These are the namespace comments for SbsSW.SwiPICs.Exceptions.</p> <p>The namespace SbsSW.SwiPICs.Exceptions provides the Exception classes to catch a prolog exception see SWI-Prolog Manual - 4.9 ISO compliant Exception handling</p> <p>Prolog exceptions are mapped to C# exceptions using the subclass PIException of Exception to represent the Prolog exception term.</p> <p>All type-conversion functions of the interface raise Prolog-compliant exceptions, providing decent error-handling support at no extra work for the programmer.</p> <p>For some commonly used exceptions, subclasses of PIException have been</p>

created to exploit both their constructors for easy creation of these exceptions as well as selective trapping in C#.

Currently, these are [PITypeException](#) and [PIDomainException](#).

[**SbsSW.SwiPICs.Streams**](#)

The namespace SbsSW.SwiPICs.Streams provides the delegates to redirect the read and write functions of the [SWI-Prolog](#) IO Streams

When [Initialize\(String\[\]\)](#) is called the *Sinput->functions.read is replaced by the .NET method 'Sread_function' and *Sinput->functions.write by 'Swrite_funktion'.

For further examples see the methods [SetStreamFunctionWrite\(PIStreamTypeDelegateStreamWriteFunction\)](#) and [SetStreamFunctionRead\(PIStreamTypeDelegateStreamReadFunction\)](#)

by Uwe Lesta, SBS-Softwaresysteme GmbH

Send comments on this topic to [Lesta at SBS-Softwaresysteme.de](mailto:Lesta@SBS-Softwaresysteme.de)