CTI OS Keywords and Enumerated Types

Keywords

The CTI OS Client Interface Library uses the Arguments structure to pass key-value pairs between the client and the server (for more information about Arguments, see Helper Classes). Throughout this document all event and method parameter lists, as well as object properties, are listed with the keywords and the types associated with those keywords.

The expected (required and optional) keywords are referred to in this document by string name. For example, the Agent's property for agent identifier is referred to as AgentID.

In addition to using the string name for a keyword, programmers can take advantage of an enumeration of keywords as well.

Note The enumeration of keywords is presently only available in the C++ CIL.

For each string keyword, a corresponding enumerated keyword exists. The enumerated keyword is the same name, preceded by the prefix “ekw”. For example, the AgentID string keyword is mapped to the enumerated keyword ekwAgentID.

Usage Example in C++:

```
Arguments& args = Arguments::CreateInstance();
args.AddItem(ekwAgentID, "22866");
args.AddItem(ekwAgentInstrument, "23901");

pAgent->Login(args);

args.Release();
```

The complete set of standard keywords used in CTI OS is available in the C++ header file “ctioskeywords.h”, located in the \Distribution\cpp\Include directory on the CTI OS toolkit media.

Java CIL Keywords

For Java CIL, the CtiOs_IKeywordIDs interface contains a list of known Java CIL CTI OS keywords. For more information, see the Java CIL Javadoc file.

.NET CIL Keywords

The Cisco.CtiOs.Util.Keywords.Enum_CtiOs enum contains the list of CTI OS keyword IDs.

Enumerated Types

CTI OS employs enumerated types to provide symbolic names for commonly recurring values:

- In C++, Visual Basic, and COM, these are presented as enumerated types.
- In Java, special interfaces are used to simulate enumerated types. For more information, see Java Interfaces, on page 2.

The complete set of enumerated types and their values are available in the following locations:

- For C++ CIL using static libraries: the complete set of enumerated types is located in the C++ header file "cilmessages.h", located in the C:\Program Files\Cisco Systems\CTIOS Client\CTIOS Toolkit\Win32 CIL\Include directory on the CTI OS toolkit media.
- For COM (Visual Basic and Visual C++): the complete set of enumerated types is located in the CTIOSClient Type Library, which is compiled into the "CTIOSClient.dll" file, located in the C:\Program Files\Cisco Systems\CTIOS Client\CTIOS Toolkit\Win32 CIL\COM Servers and Activex Controls directory on the CTI OS toolkit media.

In the Java CIL, the CTIOS_Enums interface contains the Java CIL enumerated types. For more information, see the Java CIL Javadoc file.

In the .NET CIL, the CtiOs_Enums class contains the .NET CIL enumerated types:

- For Java: To be supplied with Java package release.

Java Interfaces

The Java CIL handles the C++ CIL enums through the use of interfaces. The custom application can then either implement those interfaces and use the static data members without referencing them with the interface name first, or it can access those members through referencing. By convention, the name of the Java interface is the same as the enum tag but with the "enumCTIOS_" prefix substituted with "CtiOs_I". So for example, the following C++ CIL enum:

```
enum enumCTIOS_AgentState{
eLogin = 0,
eLogout = 1,
eNotReady = 2,
eAvailable = 3,
eTalking = 4,
eWorkNotReady = 5,
eWorkReady = 6,
eBusyOther = 7,
eReserved = 8,
eUnknown = 9,
eHold=10

 };
```

is implemented in the Java CIL as follows:

```
public interface CtiOs_IAgentState{
public static final inteLogin = 0,
eLogout = 1,
eNotReady = 2,
eAvailable = 3,
eTalking = 4,
eWorkNotReady = 5,
eWorkReady = 6,
eBusyOther = 7,
eReserved = 8,
eUnknown = 9,
eHold=10;
}
```

A Java CIL application can access those defined values in one of two ways; either by implementing the interface, as shown:

```

public class MyAgent extends CtiOsObject implements CtiOs_IAgentState
{

      ................................................


      public int MyLogin(Arguments rArguments)
     {
.................................
               //Access eLogin directly
rArguments.AddItemInt( "agentstate", eLogin );
.................................

     }
}
```

or by referencing as follows:

```
public class MyAgent extends CtiOsObject{

      ................................................
     public int MyLogin(Arguments rArguments)
     {
.................................
rArguments.AddItemInt( "agentstate", CtiOs_IAgentState.eLogin );
.................................
```

```
            }
}
```