Elisy .Net Bridge component adds .Net support to 1C:Enterprise 7.7/8.0/8.1

Main features of Elisy .Net Bridge are:
- does not require REGSVR32 and REGASM preregistration. Can be accessible by file name from 1C:Enterprise script;
- supports REGSVR32 and REGASM registration;
- supports .Net object creation by assembly file path;
- supports .Net object creation by assembly full name;
- contains ElisyNetBridge specialized .Net class containing extra functionality;

ElisyNetBridge class allows to:
- load assemblies by file path and by full name;
- create objects using constructors with up to 5 parameters;
- call methods with up to 5 parameters;
- call static methods with up to 5 parameters;
- get and set properties and fields;
- get and set static parameters;
- get and set indexers;

Elisy .Net Bridge does not support in the current version:
- Enums;
- Events subscriptions;

The attached archive contains evaluation copy of the add-in and How To topics describing the component integration and use of regular expressions in 1C:Enterprise. Registered forum users can download component from
http://www.1centerprise.com/forums/
.

How To topics are available at
Marketplace area
.

The following topics describe how 1C:Enterprise can target the .NET platform.

Elisy.NetBridge integration

How to: Create .Net object from assembly file

```
AttachAddIn("AddIn.ElisyNetBridge");
AddIn = New("AddIn.ElisyNetBridge");
net = AddIn.NewFrom("C:Program Files1cv81binElisy.NetBridge.dll",
"Elisy.ElisyNetBridge");
```

How to: Create .Net object from GAC

```
AttachAddIn("AddIn.ElisyNetBridge");
AddIn = New("AddIn.ElisyNetBridge");
message = AddIn.New("System, Version=2.0.0.0, Culture=neutral,
```

PublicKeyToken=b77a5c561934e089", "System.Net.Mail.MailMessage");

Regular expressions

Demonstrates various string operations using regular expressions classes in the .NET Framework. The following topics demonstrate the use of the .NET Framework System.Text.RegularExpressions namespace to search, parse, and modify strings.

How to: Use Regular Expressions for Simple Matching

The following code example uses regular expressions to look for exact substring matches. The search is performed by the static IsMatch method, which takes two strings as input. The first is the string to be searched, and the second is the pattern to be searched for.

```
AttachAddIn("AddIn.ElisyNetBridge");
AddIn = New("AddIn.ElisyNetBridge");
net = AddIn.New("Elisy.NetBridge, Version=0.0.0.0, Culture=neutral,
PublicKeyToken=null", "Elisy.ElisyNetBridge");

sentence = New Array();
sentence.Add("cow over the moon");
sentence.Add("Betsy the Cow");
sentence.Add("cowering in the corner");
sentence.Add("no match here");

matchStr = "cow";

For i=0 To sentence.Count()-1 Do
   Message(sentence[i]);
   If net.CallStatic2("System.Text.RegularExpressions.Regex", "IsMatch",
sentence[i], matchStr) Then
      Message(" (match for " + matchStr + " found)");
   Else
      Message(" (match for " + matchStr + " not found)");
   EndIf;
EndDo;
```

How to: Use Regular Expressions to Extract Data Fields

The following code example demonstrates the use of regular expressions to extract data from a formatted string. The following code example uses the Regex class to specify a pattern that corresponds to an e-mail address. This patter includes field identifiers that can be used to retrieve the user and host name portions of each e-mail address. The Match class is used to perform the actual pattern matching. If the given e-mail address is valid, the user name and host names are extracted and displayed.

```
AttachAddIn("AddIn.ElisyNetBridge");
AddIn = New("AddIn.ElisyNetBridge");
net = AddIn.New("Elisy.NetBridge, Version=0.0.0.0, Culture=neutral,
PublicKeyToken=null",          "Elisy.ElisyNetBridge");

address = New Array();
address.Add("jay@southridgevideo.com");
address.Add("barry@adatum.com");
address.Add("treyresearch.net");
address.Add("karen@proseware.com");

emailregex = net.New1("System.Text.RegularExpressions.Regex",
"(?<user>[^@]+)@(?<host>.+)");

For i=0 To address.Count()-1 Do
   m = net.Call1(emailregex, "Match", address[i]);
   Message(address[i]);
   If net.Get(m, "Success") Then
      groups = net.Get(m, "Groups");
      user = net.GetIndexed(groups, "Item",          "user");
      host = net.GetIndexed(groups, "Item",          "host");
      Message(" User=" + net.Get(user,          "Value"));
      Message(" Host=" + net.Get(host,          "Value"));
   Else
      Message(" (invalid email address)");
   EndIf
EndDo
```

How to: Use Regular Expressions to Rearrange Data

The following code example demonstrates how the .NET Framework regular expression support can be used to rearrange, or reformat data. The following code example uses the Regex and Match classes to extract first and last names from a string and then display these name elements in reverse order.

The Regex class is used to construct a regular expression that describes the current format of the data. The two names are assumed to be separated by a comma and can use any amount of white-space around the comma. The Match method is then used to analyze each string. If successful, first and last names are retrieved from the Match object and displayed.

```
AttachAddIn("AddIn.ElisyNetBridge");
AddIn = New("AddIn.ElisyNetBridge");
```

```
net = AddIn.New("Elisy.NetBridge, Version=0.0.0.0, Culture=neutral,
PublicKeyToken=null",          "Elisy.ElisyNetBridge");

name = New Array();
name.Add("Abolrous, Sam");
name.Add("Berg,Matt");
name.Add("Berry , Jo");
name.Add("www.contoso.com");

reg = net.New1("System.Text.RegularExpressions.Regex",
"(?<last>w*)s*,s*(?<first>w*)");

For i=0 To name.Count()-1 Do
   Message(name[i]);
   m = net.Call1(reg, "Match", name[i]);
   if net.Get(m, "Success") Then
      groups = net.Get(m, "Groups");
      first = net.GetIndexed(groups, "Item",          "first");
      last = net.GetIndexed(groups, "Item",          "last");
      Message(net.Get(first, "Value")          + " " + net.Get(last, "Value"));
   Else
      Message("(invalid)");
   EndIf;

EndDo;
```

How to: Use Regular Expressions to Search and Replace

The following code example demonstrates how the regular expression class Regex can be used to perform search and replace.

This is done with the Replace method. The version used takes two strings as input: the string to be modified, and the string to be inserted in place of the sections (if any) that match the pattern given to the Regex object.

```
AttachAddIn("AddIn.ElisyNetBridge");
   AddIn = New("AddIn.ElisyNetBridge");
   net = AddIn.New("Elisy.NetBridge, Version=0.0.0.0, Culture=neutral,
PublicKeyToken=null",          "Elisy.ElisyNetBridge");

   before = "The q43uick bro254wn f0ox ju4mped";
   Message("original : " + before);
```

```
digitRegex = net.New1("System.Text.RegularExpressions.Regex", "(?<digit>[0-9])");
after = net.Call2(digitRegex, "Replace", before, "_");

Message("1st regex : " + after);

underbarRegex = net.New1("System.Text.RegularExpressions.Regex", "_");
after2 = net.call2(underbarRegex, "Replace", after, "");
Message("2nd regex : " + after2);
```

How to: Use Regular Expressions to Validate Data Formatting

The following code example demonstrates the use of regular expressions to verify the formatting of a string. In the following code example, the string should contain a valid phone number. The following code example uses the string "d{3}-d{3}-d{4}" to indicate that each field represents a valid phone number. The "d" in the string indicates a digit, and the argument after each "d" indicates the number of digits that must be present. In this case, the number is required to be separated by dashes.

```
AttachAddIn("AddIn.ElisyNetBridge");
AddIn = New("AddIn.ElisyNetBridge");
net = AddIn.New("Elisy.NetBridge, Version=0.0.0.0, Culture=neutral,
PublicKeyToken=null", "Elisy.ElisyNetBridge");

number = New Array();
number.Add("123-456-7890");
number.Add("444-234-22450");
number.Add("690-203-6578");
number.Add("146-893-232");
number.Add("146-839-2322");
number.Add("4007-295-1111");
number.Add("407-295-1111");
number.Add("407-2-5555");

regStr = "^d{3}-d{3}-d{4}$";
For i=0 To number.Count()-1 Do
   Message(number[i]);
   If (net.CallStatic2("System.Text.RegularExpressions.Regex", "IsMatch",
number[i], regStr)) Then
      Message(" - valid");
   Else
      Message(" - invalid");
   EndIf;
EndDo;
```

How to: Parse Strings Using Regular

Expressions

The following code example demonstrates simple string parsing using the Regex class in the System.Text.RegularExpressions namespace. A string containing multiple types of word delineators is constructed. The string is then parsed using the Regex class in conjunction with the Match class. Then, each word in the sentence is displayed separately.

```
AttachAddIn("AddIn.ElisyNetBridge");
AddIn = New("AddIn.ElisyNetBridge");
net = AddIn.New("Elisy.NetBridge, Version=0.0.0.0, Culture=neutral,
PublicKeyToken=null",          "Elisy.ElisyNetBridge");

//How to: Parse Strings Using Regular Expressions
words = 0;
pattern = "[a-zA-Z]*";
Message("Pattern : " + pattern);
regex = net.New1("System.Text.RegularExpressions.Regex", pattern);
line = "one.two three:four,five six seven";

match = net.Call1(regex, "Match", line);
While net.Get(match, "Success") Do
   matchValue = net.Get(match, "Value");
   if net.Get(matchValue, "Length") >0 Then
      words = words + 1;
      Message(matchValue);
   EndIf;
   match = net.Call(match, "NextMatch");
EndDo;

Message("Number of Words : " + words);
```

Graphics operations

Demonstrates image manipulation using the Windows Software Development Kit (SDK).

The following topics demonstrate the use of the System.Drawing.Image class to perform image manipulation.

How to: Convert Image File Formats with the .NET Framework

The following code example demonstrates the System.Drawing.Image class and the

System.Drawing.Imaging..::.ImageFormat enumeration used to convert and save image files. The following code loads an image from a .jpg file and then saves it in both .png and .bmp file formats.

```
        AttachAddIn("AddIn.ElisyNetBridge");
  AddIn = New("AddIn.ElisyNetBridge");
  net = AddIn.New("Elisy.NetBridge, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null",
"Elisy.ElisyNetBridge");

  //How to: Convert Image File Formats with the .NET Framework
  drawing = net.LoadAssembly("System.Drawing, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a");
  image = net.CallStatic1("System.Drawing.Image", "FromFile", "d:SampleImage.jpg");
  pngFormat = net.GetStatic("System.Drawing.Imaging.ImageFormat", "Png");
  net.Call2(image, "Save", "d:SampleImage.png", pngFormat);
  bmpFormat = net.GetStatic("System.Drawing.Imaging.ImageFormat", "Bmp");
  net.Call2(image, "Save", "d:SampleImage.bmp", bmpFormat);
```