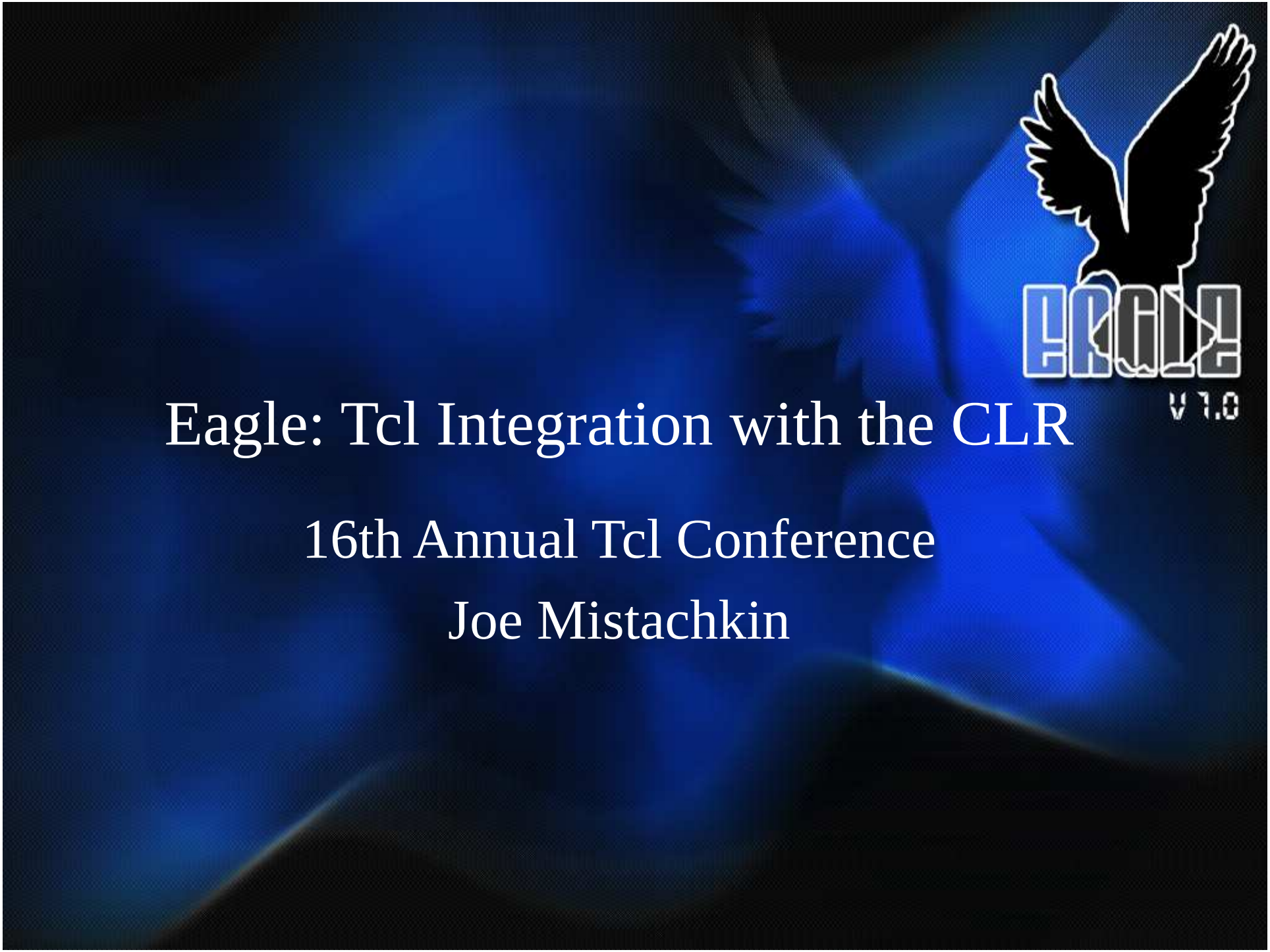

Eagle: Tcl Integration with the CLR

16th Annual Tcl Conference

Joe Mistachkin

What is Eagle?

What is Eagle?

- Eagle is an open-source implementation of Tcl written in C# for the CLR.
- Designed to be highly extensible and easily embeddable into CLR-based applications.
- Supports approximately 90% of the Tcl 8.4 core command set.

Wait a minute…

- Why integrate with Tcl if Eagle is already an implementation of the Tcl language?

Why integrate with Tcl?

- The existing Tcl integration solutions for the CLR were less than ideal.
- The possibility of using the expressive power of a Tcl-compatible language to control the integration of the Tcl language with the CLR was very compelling.
- Allows unit-testing of embedding scenarios that need to dynamically load and unload the Tcl library.

Tcl Integration Features

- Supports any build of Tcl/Tk, version 8.4 or higher, including "basekits" and "stardlls".
- Supports 32-bit and 64-bit operating systems.
- Supports worker threads with their own event loop processing.
- Supports exit handler creation and deletion.
- Supports script cancellation via TIP 285.
- Supports a unified stack trace for errors via the Eagle errorInfo variable.
- Supports error line number management via TIP 336.
- Prevents a deleted interpreter from being used.
- Prevents an interpreter with activations from being deleted via TIP 335.

Loading Tcl Automatically

```
if {![tcl ready]} then {
  #
  # This command loads the "best" Tcl
  # library available.  The exact
  # algorithm used is documented in
  # the paper.
  #
  tcl load
}
```

Loading Tcl with Version Requirements

```
if {![tcl ready]} then {
  #
  # This command loads the "best" Tcl
  # library available that meets the
  # specified version requirements.
  #
  tcl load –minimumversion 8.4 \
       -maximumversion 8.5
}
```

Loading a Specific Tcl Library

```
if {![tcl ready]} then {
  #
  # This command will skip the Tcl library
  # detection logic and simply load the
  # specified Tcl library.
  #
  tcl load {C:\full\path\to\tcl.dll}
}
```

Script Evaluation

```
set interp [tcl create]; # create a Tcl interp.

tcl eval $interp {
   #
   # This is a Tcl script.  The "eagle" command
   # here is actually a Tcl command that has
   # been implemented in managed code (in this
   # case, it simply maps to the "eval" command
   # in Eagle).
   #
   eagle debug break; # type "#go" to continue
}

tcl delete $interp; # delete the Tcl interp.
```

Using Tk in the Main Thread

```
set interp [tcl create]; # create a Tcl interp.

#
# This will load Tk into the new Tcl interp.
#
tcl eval $interp {package require Tk}

#
# Service the Tcl event loop until we are canceled.
#
after 10000 [list interp cancel]
catch {while {1} { tcl update }}

tcl delete $interp; # delete the Tcl interp.
```

Creating a Tcl Worker Thread

```
# Initialize by-ref arguments.
set result ""
set thread null

# Get the active Eagle interp.
set interp [object invoke -alias Interpreter GetActive]

# Create Tcl worker thread.
$interp -alias CreateTclThread 100 true result

# Extract the thread name from the result.
set name [$result ToString]

# Get the object for the thread we just created.
$interp -alias -flags +NonPublic GetTclThread $name \
    thread result
```

Using a Tcl Worker Thread

```
# Initialize by-ref arguments.
set result ""

# Tell the worker thread to create a Tcl interp.
$thread QueueEvent Create Immediate null true result

# Queue a script to evaluate to the worker thread.
$thread QueueEvent Evaluate Immediate \
    {package require Tk} true result

# Delete the worker thread (this will also cause the
# Tcl interp to be deleted).
$interp DeleteTclThread $name false result
```

Does it run on Mono?

(by popular demand)

- Short Answer: Yes.
- Long Answer: Yes, it runs on version 2.0 or higher. However, version 2.4 or higher is recommended due to serious bugs in earlier versions.
- Mono does not perfectly emulate the .NET Framework and still has bugs that affect Eagle (e.g. #473899, #478489, #479061, #490932).

Demonstration

Where is it?

http://eagle.to/

Questions and Answers