This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Platform SDK Developer's Guide

Managing Protocol Configuration

4/12/2025

Contents

Managing Protocol Configuration

Even after a protocol object has been created, you can still manage and update the configuration for that protocol. This article gives an overview of how to manage protocol configuration, including code samples and a list of properties that can be changed.

Managed Configuration

Starting with Platform SDK release 8.5, each protocol tracks configuration changes and applies them in any state. Some properties (such as running timer) have a deferred effect, while others are applied immediately.

The following code samples show Genesys recommendations for changing a protocol's configuration in any state. You do not need to directly set the new configuration to an Endpoint.

[+] Java Code Sample

```
// Example 1
ConnectionConfiguration cfg = protocol.getEndpoint().getConfiguration();
if (cfg instanceof ClientConnectionOptions) {
  ClientConnectionOptions options =
(ClientConnectionOptions)protocol.getEndpoint().getConfiguration();
  options.setUseAddp(true);
  options.setAddpClientTimeout(5000);
  options.setAddpServerTimeout(5000);
  options.setAddpTraceMode(AddpTraceMode.Local);
}

// Example 2
ConnectionConfiguration cfg = protocol.getEndpoint().getConfiguration();
cfg.setOption(Interceptor.PROTOCOL_NAME_KEY, AddpInterceptor.NAME);
cfg.setOption(AddpInterceptor.TIMEOUT_KEY, "5");
cfg.setOption(AddpInterceptor.REMOTE_TIMEOUT_KEY, "5");
cfg.setOption(AddpInterceptor.TRACE_KEY, "1");
```

[+] .NET Code Sample

```
// Example 1
var configuration = protocol.Endpoint.GetConfiguration() as IClientConnectionOptions;
if (configuration != null)
{
  configuration.AddpClientTimeout = 15;
  configuration.AddpServerTimeout = 20;
  configuration.AddpTraceMode = AddpTraceMode.Both;
  configuration.UseAddp = true;
}

// Example 2
var configuration = protocol.Endpoint.GetConfiguration();
if (configuration != null)
{
  configuration.SetOption(AddpInterceptor.TimeoutKey,"15");
  configuration.SetOption(AddpInterceptor.RemoteTimeoutKey, "20");
  configuration.SetOption(AddpInterceptor.TraceKey, AddpTraceMode.Both.ToString("F"));
  configuration.SetOption(CommonConnection.ProtocolNameKey,AddpInterceptor.Name);
}

// Example 3
var configuration = protocol.Endpoint.GetConfiguration();
if (configuration != null)
{
  configuration.SetOption("addp-timeout","15");
  configuration.SetOption("addp-remote-timeout", "20");
  configuration.SetOption("addp-trace", "both");
  configuration.SetOption("protocol","addp");
}
```

Managed Properties

The following tables list properties that can be changed at any time.

Properties that relate to all protocols:

Property Name	Property Type	Mnemonic Constant in Java	Mnemonic Constant in .NET
protocol	string (protocol name)	Interceptor.PROTOCOL_NAME_KEY	CommonConnection.ProtocolNameKey
addp-timeout	float (in seconds)	AddpInterceptor.TIMEOUT_KEY	AddpInterceptor.TimeoutKey
addp-remote-timeout	float (in seconds)	AddpInterceptor.REMOTE_TIMEOUT_KEY	AddpInterceptor.RemoteTimeoutKey
addp-trace	int	AddpInterceptor.TRACE_KEY	AddpInterceptor.TraceKey
string-attribute-encoding	string	Connection.STR_ATTR_ENCODING_NAME_KEY	ConnectionBase.StringAttributeEncodin

Properties that are supported by the Voice protocol:

Property Name	Property Type	Mnemonic Constant in Java	Mnemonic Constant in .NET
tspAppName	string	TServerProtocol.APP_NAME_KEY	TServerProtocol.ApplicationNameKey
tspPassword	string	TServerProtocol.PASS_KEY	TServerProtocol.PassKey
tspSwitchoverTimeout	long	TServerProtocol.SWITCHOVER_TIMEOUT_KEY	TServerProtocol.SwitchoverTimeoutKey
tspBackupReconnectInterval	long	TServerProtocol.BACKUP_RECONNECT_INTERVAL_KEY	TServerProtocol.BackupReconnectInterv

Properties that are supported by the WebMedia protocol:

Property Name	Property Type	Mnemonic Constant in Java	Mnemonic Constant in .NET
replace-illegal-unicode-chars	boolean	WebmediaChannel.OPTION_NAME_REPLACE_ILLEGAL_UNICODE_C	WebmediaChannel.OptionNameReplace
illegal-unicode-chars-replacement	string	WebmediaChannel.OPTION_NAME_ILLEGAL_UNICODE_CHARS_REPL	WebmediaChannel.OptionNameIllegal

Managing Configuration Prior to Release 8.5

For releases prior to 8.5, the configuration of an existing protocol object can still be changed. However, any configuration changes made will only take effect if the protocol object is in a "Closed" state; otherwise the changes are applied the next time that protocol is opened.

The following code examples show how Genesys recommends managing protocol configuration:

[+] Java Code Sample

```
// Example 1
ConnectionConfiguration cfg = protocol.getEndpoint().getConfiguration();
if (cfg instanceof ClientConnectionOptions) {
  ClientConnectionOptions options = (ClientConnectionOptions)cfg;
  options.setUseAddp(true);
  options.setAddpClientTimeout(5000);
  options.setAddpServerTimeout(5000);
  options.setAddpTraceMode(AddpTraceMode.Local);
  protocol.configure(cfg); // method configure is deprecated
}

// Example 2
ConnectionConfiguration cfg = protocol.getEndpoint().getConfiguration();
cfg.setOption(Interceptor.PROTOCOL_NAME_KEY, AddpInterceptor.NAME);
cfg.setOption(AddpInterceptor.TIMEOUT_KEY, "5");
cfg.setOption(AddpInterceptor.REMOTE_TIMEOUT_KEY, "5");
cfg.setOption(AddpInterceptor.TRACE_KEY, "1");
protocol.configure(cfg); // method configure is deprecated

// Example 3
ConnectionConfiguration cfg = protocol.getEndpoint().getConfiguration();
cfg.setOption("protocol", "addp");
cfg.setOption("addp-timeout", "5");
cfg.setOption("addp-remote-timeout", "5");
cfg.setOption("addp-trace", "1");
protocol.configure(cfg); // method configure is deprecated
```

[+] .NET Code Sample

```
// Example 1
var configuration = protocol.Endpoint.GetConfiguration() as IClientConnectionOptions;
if (configuration != null)
{
  configuration.UseAddp = true;
  configuration.AddpClientTimeout = 15;
  configuration.AddpServerTimeout = 20;
  configuration.AddpTraceMode = AddpTraceMode.Both;
  protocol.Configure(configuration as IConnectionConfiguration); // method Configure is
obsolete
}

// Example 2
var configuration = protocol.Endpoint.GetConfiguration();
if (configuration != null)
{
  configuration.SetOption(CommonConnection.ProtocolNameKey,AddpInterceptor.Name);;
  configuration.SetOption(AddpInterceptor.TimeoutKey,"15");
  configuration.SetOption(AddpInterceptor.RemoteTimeoutKey, "20");
  configuration.SetOption(AddpInterceptor.TraceKey, AddpTraceMode.Both.ToString("F"));
  protocol.Configure(configuration); // method Configure is obsolete
}

// Example 3
var configuration = protocol.Endpoint.GetConfiguration();
if (configuration != null)
{
  configuration.SetOption("protocol","addp");;
  configuration.SetOption("addp-timeout","15");
  configuration.SetOption("addp-remote-timeout", "20");
  configuration.SetOption("addp-trace", "both");
  protocol.Configure(configuration); // method Configure is obsolete
}
```