ibec_StartTraceSession

ibec_StartTraceSession starts a trace session using Firebird Services API and writes trace data into a file. Additional filtering of trace items is possible via callback block.

This IBEBlock function was updated in IBExpert version 2022.11.14.

Description

No additional description...

Syntax

```
  function ibec_StartTraceSession(ConnectParams : string; TraceConfig :
string; OutputFile : string;
                                FilterBlock : variant; ProgressBlock :
variant) : variant;
```

ConnectParams - list of connection params and some additional options delimited with semicolon:

- *Server* - server name including port number if necessary
- *User* - user name
- *Password* - password
- *ClientLib* - path to a client library dll
- *MaxFileSize* - maximum size of the result file, in megabytes. If not specified the result file size will be limited to 100 megabytes.
- *StopAfter* - determines duration of trace session in minutes. Default value is 60 minutes.
- *IncludeStatistics* - some statistics data (total and max per second number of processed lines and events) will be written at beginning of trace data
- *AppendToExisting* | *AppendMode* - if specified, trace data will be added to existing file (if exists)
- *ParseData* option If specified, event text will be parsed and results will be passed to filter block. In that case first input parameter of filter block will contain array of parsed data instead of single value as without ParseData option. The very first value contains plain (not parsed) event item text in both cases. To access items of parsed data array there are some constants defined in the TraceAPI namespace (see example below).
- *NoWait* option: If specified, IBEBlock will continue execution just after starting a trace session, so it is possible to start and control several trace sessions from a single block. In this case it is necessary to create a loop which will be executed while at least one session is alive (see example below). Otherwise all trace sessions will be terminated at the end of block execution.
- *SessionName* option is just to assign an user-defined name to a trace session.

TraceConfig - Firebird trace configuration

OutputFile - name of a result file which will contain trace data

FilterBlock - a callback block which performs additional filtering of trace items if necessary. This block is called for every trace item and accepts a trace item text as input parameter (*EventSource*). The

output parameter *NeedSkip* determines whether the trace item should be skipped i.e. not included into the result file.

Support of a second input parameter has beeb added both to filter and progress blocks:

- *SessionData[0] or SessionData['SessionObject']* - pointer to the trace session object
- *SessionData[1] or SessionData['SessionName']* - name of trace session
- *SessionData[2] or SessionData['SessionObject']* - internal Firebird ID of trace session

If no additional filtering is needed pass NULL or an empty string as FilterBlock value.

ProgressBlock - a callback block that is intended to display progress of tracing. It receives array of some statistical data:

- *Data[0] (or Data['LinesProcessed'])* - number of processed lines
- *Data[1] (or Data['EventsProcessed'])* - number of processed trace items/events
- *Data[2] (or Data['EventsMatched'])* - number of trace items written to result file
- *Data[3] (or Data['OutputSize'])* - current size of result file in bytes

The output parameter *Threshold* determines a delay in milliseconds before next call of *ProgressBlock*.

ibec_StartTraceSession returns NULL if a trace session is started successfully, otherwise it returns error message as a result.

TraceAPI.GetActiveTraceSessions function: By default (the Options param is NULL or contains unknown data) returns a number of active trace sessions. If the *Return=NamesList* option is specified, it will return a comma-delimited list of names of active trace sessions.

TraceAPI namespace: contains two functions and some constants. *TraceAPI.StartTraceSession* is just an alias for the *ibec_StartTraceSession* function.

Earlier implementations of the *ibec_StartTraceSession* function didn't return an error message if there was something wrong. This is now fixed.

Example

This example can be executed from within the IBExpert GUI, it doesn't freeze IBExpert.

```
execute ibeblock
as

  -- Filter block
  declare ibeblock filter_block (EventData variant, SessionData variant)
  returns (
    NeedSkip boolean)
  as
  begin
    NeedSkip = EventData[@TraceAPI.TED_EVENT_TYPE] = 'START_TRANSACTION';  -
- Event type
    if (not NeedSkip) then
```

```
      NeedSkip = EventData[@TraceAPI.TED_DURATION] <= 0;
  end;

  -- Progress block
  declare ibeblock progress_block (data variant, SessionData variant)
  returns (
    threshold integer = 1000)
  as
  begin
    threshold = 1000;
    for i = 0 to ibec_High(data) do
      data[i] = ibec_Cast(data[i], __typeString);

    s = ibec_Format('| %15.15s | %16.16s | %14.14s | %16.16s |', data[0],
data['EventsProcessed'], data['EventsMatched'], data['OutputSize']);
    ibec_ProgressEx(s, __poNoCRLF + __poReplaceLast);
  end;

begin
  sConfig = ibec_LoadFromFile('D:\temp\trace.conf');
  sOutFile = 'D:\Temp\tracedata.txt';
  sServer = 'avx-dell/3053';
  sClientLib = '"D:\Projects_5\IBExpert\ClientLibs\fbclient_30.dll"';


  Res = @TraceAPI.StartTraceSession('Server=' || sServer || ';
                                    User=SYSDBA; Password=masterkey;
                                    ClientLib=' || sClientLib || ';
                                    MaxFileSize=10; StopAfter=10;
IncludeStatistics; ParseData; NoWait',
                                  :sConfig,
                                  :sOutFile,
                                  filter_block,
                                  progress_block);

  if (Res is not null) then  -- Error caused
    ibec_ShowMessage('ERROR: ' || Res);
  else
  begin
    iTime1 = ibec_GetTickCount();
    ibec_Progress('Trace session started');
    ibec_Progress('+-----------------+-----------------+----------------+--
----------------+');
    ibec_Progress('| Lines processed | Events processed | Events matched |
Output file size |');
    ibec_Progress('+-----------------+-----------------+----------------+--
----------------+');
  end;

  while (@TraceAPI.GetActiveTraceSessions('Return=NamesList') <> '') do
  --while (@TraceAPI.GetActiveTraceSessions('') > 0) do
```

```
  begin
    -- ibec_Pause should be used instead of ibec_Sleep here
    ibec_Pause(2000);
  end;

  iTime2 = ibec_GetTickCount();
  sSessionTime = ibec_FormatFloat('0.00', (iTime2 - iTime1) / 1000 / 60);

  ibec_Progress('');
  ibec_Progress('+-----------------+-----------------+---------------+----
--------------+');
  ibec_Progress('Session was active ' || sSessionTime || ' minutes');
  ibec_Progress('That''s all, folks!');
end;
```

Further information and examples here: White Paper: ibec_StartTraceSession