

REVCAPI Equates

```
compile Insert REVCAPI_EQUATES

// CreateEngine Constants
equ CREATE_ENGINE_OPEN_EXISTING$ to 0x000
equ CREATE_ENGINE_CREATE_NEW$ to 0x001
equ CREATE_ENGINE_OPEN_ALWAYS$ to 0x002
equ CREATE_ENGINE_INDEXER$ to 0x010
equ CREATE_ENGINE_WAIT_ON_CLOSE$ to 0x020
equ UNPROCESSED$ to 0 ;*Server has not begun request.
equ PROCESSING$ to 1 ;* Server is processing request.
equ DATA_AVAILABLE$ to 2 ;* Server has data available.
equ COMPLETED$ to 3 ;* Server has completed request, status information is available.
equ PROC_ERROR$ to 4 ;* Server process failed, status information is available.
equ INFO_AVAILABLE$ to 5 ;* Server has intermediate status information available.
equ INFO_REQUEST$ to 10 ;* Server is requesting information from client.

declare function CreateEngine ;* Error = CreateEngine(Engine, ServerSpec, DatabaseName, Flags,
ShutdownSessions)
declare function CreateQueue ;* Error = CreateQueue(Queue, Engine, QueueName, DatabaseName, UserName, Password)
declare function CreateRequest ;* Error = CreateRequest(Request, Queue, Script, Arg1, Arg2, ...)
declare function PollForReply ;* Error = PollForReply(Request, Status[, Reply])
declare function WaitForReply ;* Error = WaitForReply(Request, Status[, Reply])
declare function GetReply ;* Error = GetReply(Request, Reply)
declare function SendResponse ;* Error = SendResponse(Request, Response)
declare function GetStatusText ;* Error = GetStatusText(Request, Delimiter, Text)
declare function CloseRequest ;* Error = CloseRequest(Request)
declare subroutine CloseRequest ;* Error = CloseRequest(Request)
declare function CallSubroutine ;* Error = CallSubroutine(Queue, Procedure, Arg1, Arg2, ...)
declare function CallFunction ;* Error = CallFunction(Queue, ReturnValue, Procedure, Arg1, Arg2,...)
declare function CloseQueue ;* Error = CloseQueue(Queue)
declare subroutine CloseQueue
declare function CloseEngine ;* Error = CloseEngine(Engine)
declare subroutine CloseEngine
```