Using VEGA TCP/IP in CA-IDMS Applications

Contents

- Introduction to TCP/IP
- VEGA TCP/IP for CA-IDMS
- Prerequisite Systems Software
- Types of Applications
- Listener
- Supported API

Contents

- Typical Client Application Logic
- Typical Server Application Logic
- API Syntax and Parameters
- Software Architecture
- Interactive Test Tool
- VEGA TCP/IP for BATCH

Contents

- Example 1: DC-COBOL Server Application
- Example 2: VG-IDMS/Web Server
- Summary

Introduction to TCP/IP

- Address
  - 32-bit binary number
  - dotted decimal notation, e.g. 192.1.10.1
  - Domain name resolution (What's the address of www.vegasoft.com?)
- Port
  - 16-bit binary number
  - identifies a specific service or application

Introduction to TCP/IP

- Service
  - FTP, Telnet, SMTP, HTTP
  - “well known ports”
- Socket
  - application programming interface
  - semantically quite similar to LU6.2
  - client sockets initiate connections
  - server sockets wait for and accept connections

VEGA TCP/IP for CA-IDMS

- Runs under CA-IDMS/DC/UCF 10.21 or later (DC is not required)
- Provides CICS sockets compatible API
- Can be used in
  - CA-ADS
  - DC-COBOL, Assembler, PL/1
  - BATCH-COBOL
    - Original CICS Socket API (Vegasoft implementation)
    - Socket Extended API (IBM implementation)

VEGA TCP/IP for CA-IDMS

- Cross-address space communication uses MVS subtasks
- Control tasks available to
  - (re)start TCP/IP interface
    - cycling CV is not needed
  - stop TCP/IP interface
  - monitor system activity

VEGA TCP/IP for CA-IDMS

- Proprietary Logical Terminal Architecture
- Only tasks and programs defined in Sysgen
- ASCII/EBCDIC conversion API is included

Prerequisite systems software

- TCP/IP for MVS (or OS/390 V2R5 or later)
  - IBM product (VM->MVS)
  - Own address space
  - In OS/390 TCP/IP resides in the Unix box
  - C functions
  - Inter-User Communication Vehicle (IUCV) macros
  - CICS Sockets
    - Original CICS API
    - Socket Extended API (batch also in TCP/IP V3R1)

Prerequisite systems software

- Interlink's TCPAccess 4.1 or later
  - IBM compatible
  - Supports IUCV

Types of applications

- Client applications
  - examples
    - a CA-ADS application talks to an NT application
    - a CA-IDMS application talks to another CA-IDMS application

Types of applications

- Server applications
  - examples
    - a CA-IDMS task is started by a Windows application
    - FTP
- TCP/IP Listener is included

Listener

- Needed for Server applications
- Listens to a predefined port
  - the port is defined in the Options Table
- Starts an application task
  - first 8 bytes (ASCII or EBCDIC) of the first message define the task code
  - understands HTTP
- Signon exit (R12 or later)
- Error logging

Supported API

- Original Socket API
  - parameter list compatible with the CICS EZACICAL API
- Socket Extended API
  - object code compatible with IBM TCP/IP V3R1 Socket Extended Call interface for COBOL and Assembler
- CA-ADS applications use LINK PROGRAM interface with the same parameter list

Supported API

- Accept
- Bind
- Close
- Connect
- Endapi (extension)
- Fcntl
- Getclientid
- Gethostbyaddress (x)
- Gethostbyname (x)
- Gethostid
- Gethostname
- Getpeername
- Getsockname
- Getsockopt
- Givesocket
- Initapi
- Ioctl
- Listen
- Read
- Read-all (extension) (x)
- Recvfrom
- Select
- Send
- Sendto
- Setsockopt
- Shutdown
- Socket
- Takesocket
- Termapi (x)
- Write
- Write-all (extension) (x)

(x) = Socket Extended API

Client Application Example

- Initapi
  - create a VEGA TCP/IP thread
- Socket
  - obtain a socket descriptor
- Connect
  - establish a connection to the remote server
- Write
  - send the first message to the server

Client Application Example (continues)

- One or more Reads and/or Writes
  - perform conversation with the Server
- Close
  - end the conversation and release the socket descriptor
- Endapi/Termapi
  - terminate the TCP/IP thread initiated by the Initapi

Server Application Example

- MTCPPARM
  - get the socket number, subtask name and address space id from the VEGA Listener
- Initapi
  - connect the application to the TCP/IP systems software and create a VEGA TCP/IP thread
- Takesocket
  - obtain the socket descriptor from the VEGA Listener based on the MTCPPARM call

Server Application Example (continues)

- Read
  - get data from the Client
- One or more Reads and/or Writes
  - perform conversation with the Client
- Close
  - end the conversation and release the socket descriptor
- Endapi/Termapi
  - terminate the TCP/IP thread initiated by the Initapi

Original API Syntax

- COBOL
  - CALL ’MTCPHLI’ USING TOKEN, COMMAND, P1,...,Pn, ERRNO, RETCODE
- CA-ADS
  - LINK PROGRAM ’MTCPHLIA’ USING (TOKEN, COMMAND, P1,...,Pn, ERRNO, RETCODE)
  - parameters are stored as CA-IDD records during the installation

Original API Parameters

- TOKEN

Field	Picture	Description	Value
WTCPTOK-EYE	X(16)	Eye catch	TCPIPIUCVSTREAMS
WTCPTOK-PGM	X(8)	Dialog/program name	
WTCPTOK-RET	S9(8) COMP	Extended reason code	
FILLER	X(116)	Reserved for the system	

Original API Parameters (continues)

- COMMAND
  - Binary half word with values from 1 to 41 identifying the socket call
- P1, … Pn
  - a command specific list of socket call parameters
- ERRNO
  - If RETCODE is negative, this contains an error number

Original API Parameters (continues)

- RETCODE
  - A full word binary variable which contains the code returned by the API call

Extended API Syntax

- COBOL
  - CALL 'EZASOKET' USING SOC-FUNCTION, (P1,...,Pn), ERRNO, RETCODE
- CA-ADS
  - LINK PROGRAM 'MTCPEZAA' USING (SOC-FUNCTION,(P1,...,Pn),ERRNO, RETCODE
  - parameters are stored as CA-IDD records during the installation

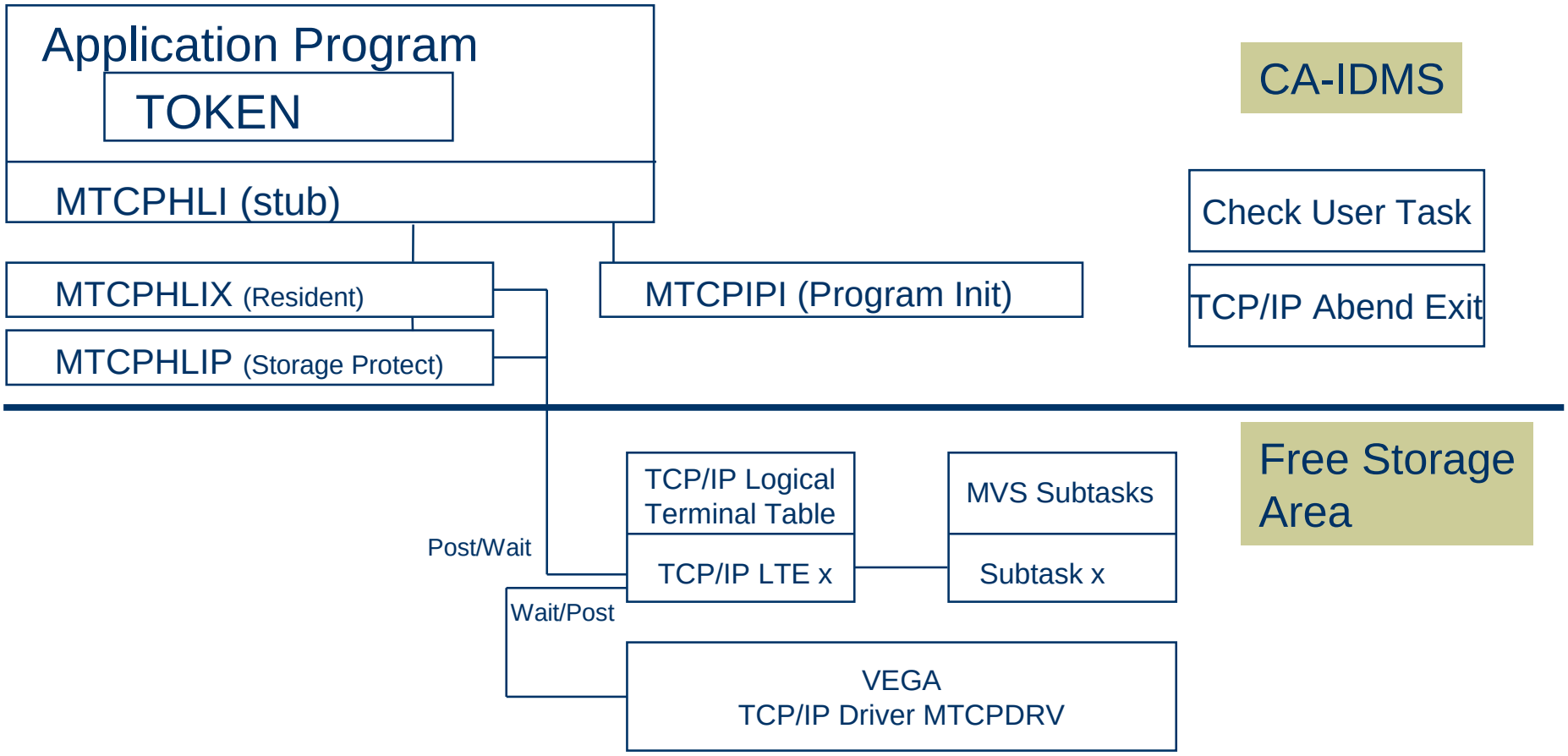
Extended API Parameters

- SOC-FUNCTION
  - A 16-byte character field identifying the socket call
- Pi
  - The socket call parameter(s)
- ERRNO
  - If RETCODE is negative, this contains an error number

Extended API Parameters

- RETCODE
  - A full word binary variable which contains the code returned by the API call

Software Architecture



Interactive Test Tool

- CA-ADS Client/Server dialogs to test installation

```
 VEGASOFT OY                VEGA                        V2R6
                            TCP/IP TEST DIALOG
 ------------------------------------------------------------------------------
           RETCODE   ERRNO      EBCDIC-ASCII-EBCDIC CONVERSION:
 INITAPI         0      49
 SOCKET          0       0
 CONNECT         0       0      PORT:   000  ADDRESS: 192 086 113 014
 WRITE           0       0      ENTER THE WRITE DATA IN 4 LINES BELOW
TCPSERV 00XXXXXXXXXXXXXXXX



 READ            0       0




 CLOSE           0       0
 ENDAPI          0       0

 ------------------------------------------------------------------------------
PRESS ENTER TO EXECUTE
```

VEGA TCP/IP for BATCH

- Compatible at object level with the TCP/IP for CA-IDMS implementation
- Listener is not available

DC-COBOL Server Example (1/6)

```
 PROCEDURE DIVISION.
  MAIN-LINE.
*GET THE SOCKET NUMBER FROM THE LISTENER
      TRANSFER CONTROL 'MTCPPARM' LINK USING WTCPPARM.
      PERFORM IDMS-STATUS.

*INITAPI
      MOVE 'INITAPI' TO LOG-FUNCTION.
      CALL 'MTCPHLI' USING  WTCPTOKEN  CMD-INITAPI
                            IDENT MSOCK API ST FZERO MAXS RC.
      PERFORM TCPIP-STATUS.
```

DC-COBOL Server Example (2/6)

```
*TAKESOCKET
    MOVE WPARM-LSTN-NAME         TO CLIANAME.
    MOVE WPARM-LSTN-SUBTASK      TO CLISTASK.
    MOVE WPARM-GIVE-TAKE-SOCKET  TO L-DESC.
    MOVE 'TAKESOCKET' TO LOG-FUNCTION.
    CALL 'MTCPHLI' USING  WTCPTOKEN  CMD-TAKESOCKET
                          HZERO CLIENTID L-DESC SOCK
                          ERRNO RC.
    PERFORM TCPIP-STATUS.
    MOVE  RC TO S.
```

DC-COBOL Server Example (3/6)

```
*TCP/IP READ HEADER FROM CLIENT
     MOVE LENGTH OF FROM-HDR TO NBYTE.
     MOVE 1         TO IND.
     MOVE 'READ' TO  LOG-FUNCTION.
     PERFORM UNTIL (NBYTE < 1)
        CALL 'MTCPHLI' USING  WTCPTOKEN  CMD-READ
                              S DZERO NBYTE FROM-HDR-X (IND)
                              ERRNO RC
        PERFORM TCPIP-STATUS
        SUBTRACT RC FROM NBYTE
        ADD      RC TO   IND
     END-PERFORM

*NOTE: THE READ-ALL FUNCTION IN THE SOCKET EXTENDED API
       CONTAINS THE FUNCTIONALITY OF THE ABOVE LOOP STRUCTURE
```

DC-COBOL Server Example (4/6)

```
*TCP/IP READ DATA   FROM CLIENT
     MOVE FROM-LEN TO NBYTE.
     MOVE 1        TO IND.
     MOVE 'READ' TO  LOG-FUNCTION.
     PERFORM UNTIL (NBYTE < 1)
        CALL 'MTCPHLI' USING  WTCPTOKEN  CMD-READ
                              S DZERO NBYTE FROM-DATA (IND)
                              ERRNO RC
        PERFORM TCPIP-STATUS
        SUBTRACT RC FROM NBYTE
        ADD      RC TO   IND
     END-PERFORM

*APPLICATION LOGIC
    ...
```

DC-COBOL Server Example (5/6)

```
*TCP/IP WRITE TO CLIENT (APPLICATION DATA)
     MOVE TO-LEN TO NBYTE.
     ADD  LENGTH OF TO-LEN TO  NBYTE.
     MOVE  1        TO IND.
     MOVE 'WRITE'  TO LOG-FUNCTION.
     PERFORM UNTIL (NBYTE < 1)
        CALL 'MTCPHLI' USING WTCPTOKEN  CMD-WRITE
                             S NBYTE FZERO SZERO TO-HDR-X (IND)
                             ERRNO RC
        PERFORM TCPIP-STATUS
        SUBTRACT RC FROM NBYTE
        ADD      RC TO   IND
     END-PERFORM
*NOTE: THE WRITE-ALL FUNCTION IN THE SOCKET EXTENDED API
       CONTAINS THE FUNCTIONALITY OF THE ABOVE LOOP STRUCTURE
```

DC-COBOL Server Example (6/6)

```
*TCP/IP CLOSE SOCKET
     MOVE 'CLOSE'  TO LOG-FUNCTION.
     CALL 'MTCPHLI' USING  WTCPTOKEN  CMD-CLOSE
                           S DZERO
                           ERRNO RC.
     PERFORM TCPIP-STATUS.

*ENDAPI
     MOVE 'ENDAPI' TO LOG-FUNCTION.
     CALL 'MTCPHLI' USING  WTCPTOKEN  CMD-ENDAPI
                           ERRNO RC.
     PERFORM TCPIP-STATUS.
     DC RETURN.
```

VG-IDMS/Web Server

- Runs under CA-IDMS/DC/UCF (DC is not required)
- Started by the Listener
- Reads/Writes TCP/IP messages
- Starts an application based on URL
- Processes query variables

VG-IDMS/Web Server

- HTML pages can be stored in CA-IDD
- Supports Basic authentication
- SSL indirectly supported with firewalls
  - secured connection between browser and firewall
  - unsecured connection between firewall and server
- Understands SQL
  - access to CA-IDMS, DB2 and ODBC
- Performs ASCII/EBCDIC conversion

VG-IDMS/Web Server

- HTML pages can contain
  - Standard HTML
  - JAVA Scripts, JAVA Applets and other active components
- Text format components can be stored as CA-IDD modules
- Binary components can be stored as CA-IDD load modules

VG-IDMS/Web Server

- Built-in functions for CA-ADS
  - HTTPHVAR
    - Gets the request header field
  - HTTPDVAR
    - Gets a query variable from an incoming message
    - Data type parsing and checking
    - MOVE HTTPDVAR(WWWWPTR,’CUSTID’,1,CUST-ID) TO CUST-ID

VG-IDMS/Web Server

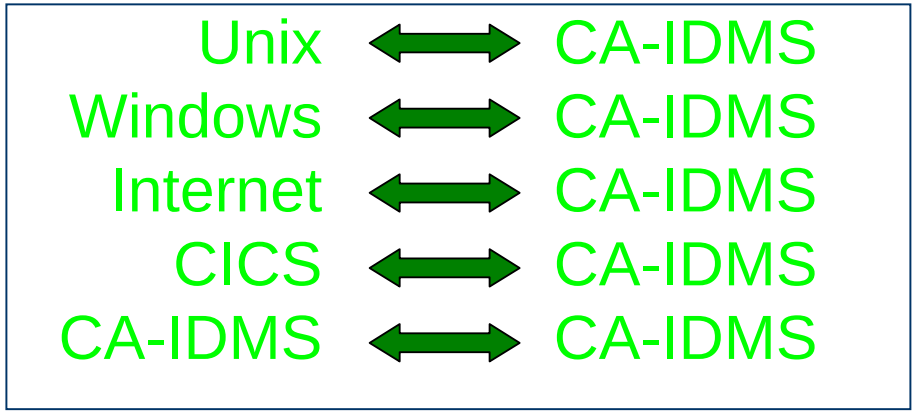
- Built-in functions for CA-ADS
  - HTMLPAGE
    - creates an HTML page into scratch area
    - page layout can be stored as a CA-IDD module
    - page layout can contain replaceable parameters
    - supports basic editing
    - MOVE HTMLPAGE (WWWWPTR, ’CUSTPAGE’, 1,’CUSTID’,CUST-ID,’CUSTNAME’,CUST-NAME, ’CUSTBAL’, CUST-BALANCE) TO WWWWHTTP-STATUS

VG-IDMS/Web Server

- New Features
  - Map Converter
    - automatically converts MAPC (OLM) maps to HTML forms
    - automatically generates CA-ADS process code to retrieve and send generated HTML forms
  - SQL Support
    - can access CA-IDMS, DB2 and ODBC

Summary

- TCP/IP now available for CA-IDMS as 3rd party products
- Where to use
  - in Client/Server applications between any platform supporting TCP/IP Sockets



End of Presentation

See
www.vegasoft.com
for
international
distributors.