



Using VEGA MQSeries in CA-IDMS Applications

VEGA50FT



Contents

- ◆ Introduction to MQSeries
- ◆ VEGA MQSeries for CA-IDMS
- ◆ Prerequisite Systems Software
- ◆ Types of Applications
- ◆ Trigger Monitor
- ◆ Supported API



Contents

- ◆ Typical Asynchronous Client Application Logic
- ◆ Typical Synchronous Server Application Logic
- ◆ API Syntax and Parameters
- ◆ Software Architecture



Contents



- ◆ Example: CA-ADS Server Application
- ◆ Utility Calls
- ◆ Online Tools
- ◆ Interactive Test Tool
- ◆ Consistent Update Method **New!**
- ◆ Summary



Introduction to MQSeries

- ◆ Messages
- ◆ Queues
- ◆ Queue manager
- ◆ Time independent, assured message delivery



Introduction to MQSeries

- ◆ Asynchronous
- ◆ Multi-platform
 - most major IBM and non IBM platforms
- ◆ MQI
 - a single multi-platform API



VEGA MQSeries for CA-IDMS

- ◆ Runs under CA-IDMS/DC/UCF 10.21 or later (DC is not required)
- ◆ MQI compatible API (100%)
- ◆ Can be used in
 - CA-ADS
 - DC-COBOL, Assembler and PL/1
- ◆ Cross-address space communication uses MVS subtasks



VEGA MQSeries for CA-IDMS

- ◆ Control tasks available to
 - (re)start MQSeries interface
 - cycling CV is not needed
 - stop MQSeries interface
 - monitor system activity
- ◆ proprietary Logical Terminal Architecture
- ◆ only tasks and programs defined in Sysgen



Prerequisite systems software

- ◆ MQSeries for MVS
 - IBM product
 - Own address space
 - MQI API



Types of applications

- ◆ Client applications

- ◆ Examples

- a CA-ADS application sends messages to an NT application
- a CA-IDMS application sends messages to another CA-IDMS application



Types of applications

- ◆ Server applications

- Example

- a CA IDMS task is triggered by messages from a Unix application

- ◆ Trigger monitor is included

Trigger Monitor

- ◆ Needed for Server applications
- ◆ Reads the given initiation queue
 - CA IDD module XQSE INITIATION QUEUES contains initiation queue names
- ◆ Starts server task (mqtm_applid)
- ◆ Automatic/manual start
- ◆ Automatic/manual stop



Trigger Monitor

- ◆ Signon exit
 - CA-IDMS R12 or later
 - current user id and password
 - new password
- ◆ Multiple trigger monitors are supported
- ◆ MQSeries (re)start allowed w/o operator intervention



Supported API



- ◆ MQBACK
- ◆ MQCLOSE
- ◆ MQCMIT
- ◆ MQCONN
- ◆ MQDISC
- ◆ MQGET
- ◆ MQINQ
- ◆ MQOPEN
- ◆ MQPUT
- ◆ MQPUT1
- ◆ MQSET



Asynchronous Client Example

- ◆ MQOPEN
 - Open an output queue and get the object handle.
- ◆ MQPUT
 - Put a message into the queue.
- ◆ MQDISC
 - Disconnect with implicit queue close and commit.

Synchronous Server Example

◆ MQSEPARM

- Get MQTM structure from the VEGA Trigger Monitor.

◆ MQOPEN

- Open an input queue MQTM-QNAME and get the object handle.

◆ MQGET

- Get a message from the client and perform database processing.

Synchronous Server Example (continues)

◆ MQOPEN

- Open an output queue MQMD-REPLYTOQ and get the object handle.

◆ MQPUT...

- Put one or more messages into the reply queue.

◆ MQDISC

- Disconnect with implicit queue close and commit.

API Syntax

◆ COBOL

- CALL '`<mqifunc>`' USING P1,...,Pn

◆ CA-ADS

- MOVE `<mqifunc>` TO MQFN-FUNC.
- LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV, P1,...,Pn)
- parameters are stored as CA-IDD records during the installation

API Parameters

- ◆ <mqifunc>
 - Identifies the MQI call
 - MQBACK, MQCLOSE, MQCMIT, MQCONN, MQDISC, MQGET, MQINQ, MQOPEN, MQPUT, MQPUT1 or MQSET
- ◆ Pi
 - MQI call parameters

API Parameters (continues)

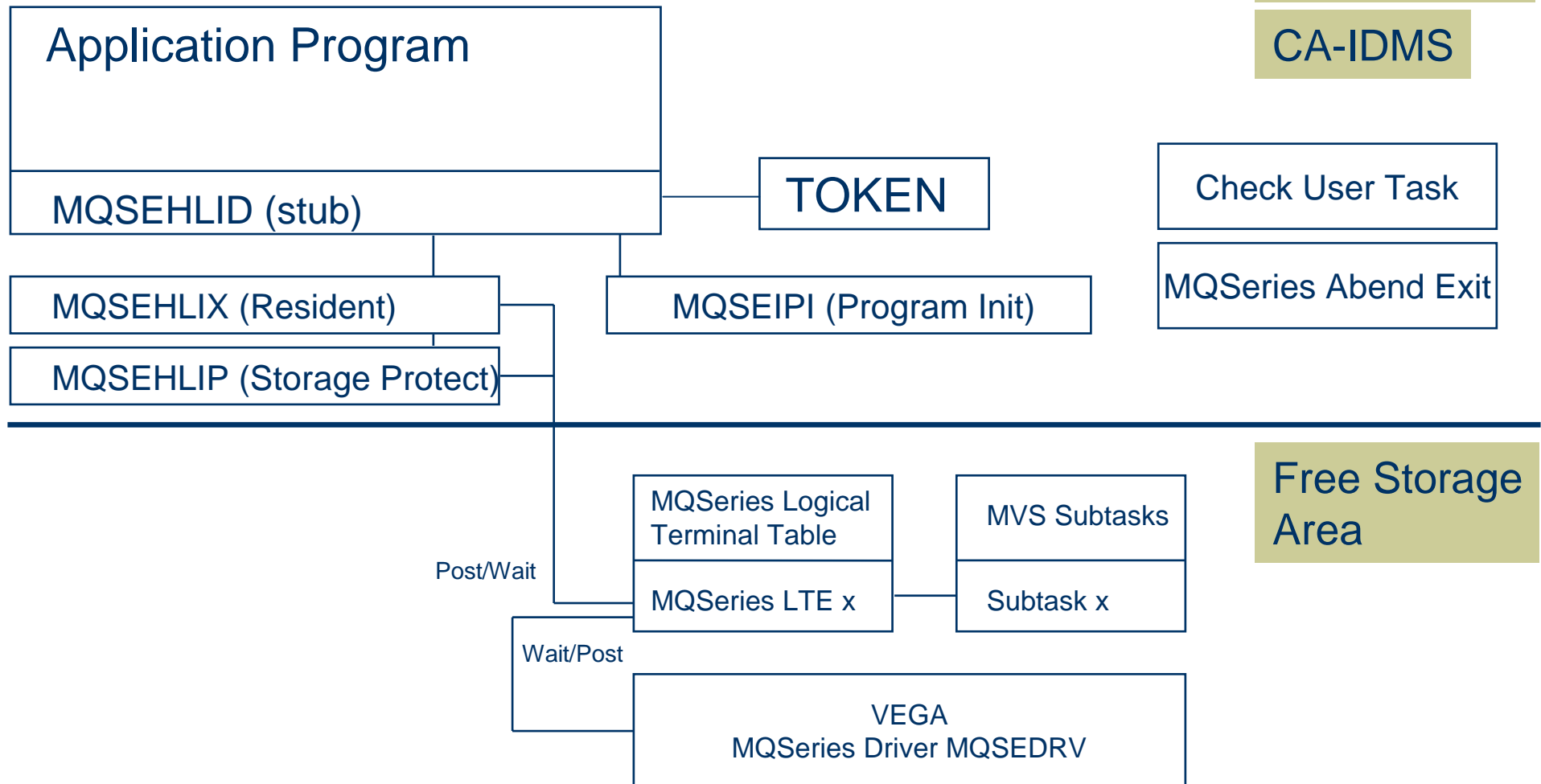
◆ WQSE-CMQFNV

- The CA-IDD record that contains the MQI function element MQFN-FUNC.

◆ WQSECNST

- The CA-IDD record that contains the <mqifunc> elements
 - MQBACK, MQCLOSE, MQCMIT, MQCONN, MQDISC, MQGET, MQINQ, MQOPEN, MQPUT, MQPUT1 and MQSET.

Software Architecture



CA-ADS Server Example (1/5)

```
!READ TRIGGER MESSAGE
LINK PROGRAM 'MQSEPARM' USING (WQSE-CMQTMV).

!OPEN INPUT QUEUE
MOVE MQTM-QNAME TO MQOD-OBJECTNAME.
MOVE MQOPEN TO MQFN-FUNC.
MOVE MQOO-INPUT-SHARED TO MQOPT-OPTIONS-OPEN.
LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV
    WQSE-CMQHCV WQSE-CMQODV,WQSE-CMQOPTV-OPEN WQSE-CMQHOV-GET
    WQSE-CMQCCV WQSE-CMQRCV).
```

CA-ADS Server Example (2/5)

!GET

MOVE MQGET TO MQFN-FUNC.

MOVE 320 TO MQBFL-BUFFERLENGTH-GET.

**LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV
WQSE-CMQHCV WQSE-CMQHOV-GET WQSE-CMQMDV
WQSE-CMQGMOM WQSE-CMQBFLV-GET WQSETST1G WQSE-CMQDLV
WQSE-CMQCCV WQSE-CMQRCV).**

!CLOSE

MOVE MQCLOSE TO MQFN-FUNC.

**LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV
WQSE-CMQHCV WQSE-CMQHOV-GET WQSE-CMQOPTV-CLOSE
WQSE-CMQCCV WQSE-CMQRCV).**

CA-ADS Server Example (3/5)

```
!OPEN FOR OUTPUT
MOVE MQOPEN TO MQFN-FUNC.
MOVE MQOO-OUTPUT TO MQOPT-OPTIONS-OPEN.
MOVE MQMD-REPLYTOQ TO MQOD-OBJECTNAME.
LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV
WQSE-CMQHCV WQSE-CMQODV WQSE-CMQOPTV-OPEN WQSE-CMQHOV-PUT
WQSE-CMQCCV WQSE-CMQRCV).
```

CA-ADS Server Example (4/5)

```
!PUT
MOVE MQPUT TO MQFN-FUNC.
MOVE 320 TO MQBFL-BUFFERLENGTH-PUT.
ACCEPT USER ID INTO WTST1P-WQSEBUFF.
MOVE CON(SUBS(WTST1P-WQSEBUFF,1,8),WTST1G-WQSEBUFF) TO
WTST1P-WQSEBUFF.
LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV
WQSE-CMQHCV WQSE-CMQHOV-PUT WQSE-CMQMDV
WQSE-CMQPMOM WQSE-CMQBFLV-PUT WQSETST1P
WQSE-CMQCCV WQSE-CMQRCV).
```

CA-ADS Server Example (5/5)

!CLOSE

MOVE MQCLOSE TO MQFN-FUNC.

**LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV
WQSE-CMQHCV WQSE-CMQHOV-PUT WQSE-CMQOPTV-CLOSE
WQSE-CMQCCV WQSE-CMQRCV).**

!COMMIT

MOVE MQCMIT TO MQFN-FUNC.

**LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV
WQSE-CMQHCV
WQSE-CMQCCV WQSE-CMQRCV).**

!DISCONNECT

MOVE MQDISC TO MQFN-FUNC.

**LINK PROGRAM 'MQSEHLIA' USING (WQSE-CMQFNV
WQSE-CMQHCV
WQSE-CMQCCV WQSE-CMQRCV).**

LEAVE ADS.



Utility Calls

- ◆ Explain MQI return and reason codes
- ◆ COBOL
 - TRANSFER 'MQSEXPL' USING CompCode, Reason, Expl LINK
- ◆ ADS
 - LINK PROGRAM 'MQSEXPL' USING (CompCode, Reason, Expl).

Utility Call Parameters

Parameter/ CA-IDD record	Picture	Description	Value
CompCode WQSE-CMQCCV	S9(9) BINARY	Completion code	
Reason WQSE-CMQRCV	S9(9) BINARY	Reason code	
Expl WQSEXPL1	GROUP	Explanation	
CompCodeDesc	X(12)	Description of the completion code	Completion code constant text MQCC_*
ReasonDesc	X(40)	Description of the reason code	Reason code constant text MQRC_*

Online Tools

(C) Vegasoft Oy

V E G A

V2R6.6 Tape V00501

MQSeries Menu

*** Please select a Response ***

—	MQSTEST		MQSeries Standard Test
—	MQSTESTC		MQSeries Conversational Test
—	MQSATTR		MQSeries INQ/SET Attributes
—	MQSLIST		MQSeries Connection List
—	MQSRLIST		MQSeries Replay Status
—	SR2MOFF		MQSeries Connection Close
—	SR2MOFFW		MQSeries Connect Close,wait
—	MENU	(PF3)	Previous Menu
—	HELP	(PF1)	Valid Responses
—	QUIT	(PA2)	Exit Application

Response:

Data:



Interactive Test Tool

- ◆ All MQI functions and parameters
- ◆ All MQ structures
- ◆ Standard mode and Conversational mode
- ◆ Explain function
 - mqi completion and reason codes
 - can be called from applications
- ◆ Restrictions
 - character format input with 320 byte I/O area

Interactive Test Tool (continues)

```
(C) Vegasoft Oy          VG-IDMS Gateway          V2R6
                        MQSeries Test Tool

-----
S Func   CC Reason                                     Object name
S Open   0      0 Options:                             16 Obj desc  PF5
S Put(1) 0      0 Length :                             0 Msg desc  PF9 Put message options PF10

S Close  0      0 Options:                             0      Object name
S Open   0      0 Options:                             1 Obj desc  PF6
N Get    0      0 Length :                             0 Msg desc  PF11 Get message options PF12

S Close  0      0 Options:                             0 -----
S Cmit   0      0                                     ! Special values for column S are:      !
  Back   0      0                                     ! 1 Put: MQPUT1                          !
S Disc   0      0                                     ! N Get: MQGET with MQMI_NONE&MQCI_NONE!

-----
Enter = Execute selected (S) functions, PF2 = CC/Reason (cursor), PF3 = Menu
READY
```



Consistent Update Method

- ◆ Keeps CA-IDMS database and MQSeries queue contents in sync
- ◆ MQSeries is first committed, then CA-IDMS
 - data is out of sync, if task or system abend occurs before CA-IDMS commit
- ◆ Replay of CA-IDMS updates
 - optional CA-IDMS database program can redo CA IDMS updates in case of a task or system abend



Consistent Update Method

- ◆ Following components are used to accomplish consistent updates:
 - MQSeries and CA IDMS synchronization queues
 - CA IDMS database application program
 - VEGA commit program
 - VEGA replay task



Summary

- ◆ MQSeries now available for CA-IDMS as 3rd party products
- ◆ Consistent Update Method helps keeping data in sync
- ◆ Where to use
 - in Client/Server applications with any platform supporting MQSeries



End of Presentation

See
www.vegasoft.com
for
international
distributors.