

www.mbfoster.com

Forging the Future



MBF-UDALink Driver

2 Phase Commit, RPC and SSL

Presented by John Middelveen
Technical Mgr. Core Product Development
MBFoster

MB Foster



John Middelveen

Technical Manager

Core Product Development

- MBF-UDALink ODBC Driver & ODBCLink/SE maintenance and enhancements
- 15 years in Canadian Military
- Degree in Electrical Engineering
- Programmer/Analyst, Message Switching Network, National Defence Headquarters



- **Every Minute**
- **Every Day**
- **Every Where**
- **Someone is using an MB Foster Solution**

Data Access

Data Marts

MBF-UDALink

**Cross-Platform
Integration**

**Client-Server &
Web Access**

**Report
Warehousing**

**Support
Services &
Education**

Delivery Solutions

**B2B
E-Commerce**

**EDI
Translation/Map**

**XML
Messaging**

FORGING THE FUTURE





What are our goals?

- Expand your revenue opportunities
- Reduce your operating expenses
- Maximize your working capital
- Preserve your investment
- Incrementally increase your profits, where we can, as fast as we can

Our Credentials

MB Foster • Forging the Future



- Twenty-four years of Hewlett-Packard knowledge and expertise.
- Trusted HP/CSY Partner
- Microsoft Certified Solutions Partner
- IBM Certified Partner
- Trusted HP Advisory Group Member
- Oracle Development Partnership
- Member Software Support Professional Assoc. (SSPA)
- Member, SIG Image Advisory Committee



MB Foster Strategic Partners



i n v e n t

oracle

IBM

Microsoft®

CERTIFIED

Partner



Overview

- **Why Transactions**
- **What are Transactions**
- **2 Phase Commit**
- **MS Distributed Transaction Coordinator**
- **Using Transactions with MBF UDALink**



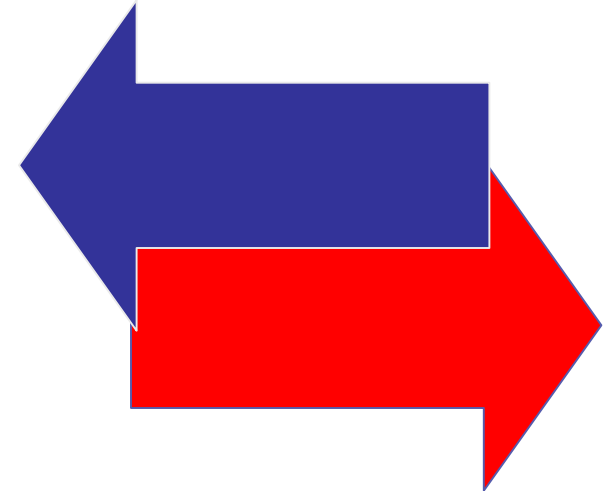
Why Transactions

- Accuracy of information while maintaining good speed– distinct business differentiator
- Computing environments have a high degree of connectivity
- Applications expected to data integrity and consistency



What are Transactions

- Action(s) that transform the system from one consistent state to another
- Building blocks required to build robust applications
- A 'Distributed' Transaction spans two or more Computer Systems





Transaction ACID Properties

- ACID
 - = Atomicity
 - = Consistency
 - = Isolation
 - = Durability



ACID

- Atomicity
 - = All or nothing property of a transaction
- Consistency
 - = Transaction transform the system from one consistent state to another



ACID

- Isolation
 - = Each transactions executes like it is the only one although there may be many
- Durability
 - = Once committed, transactions are durable



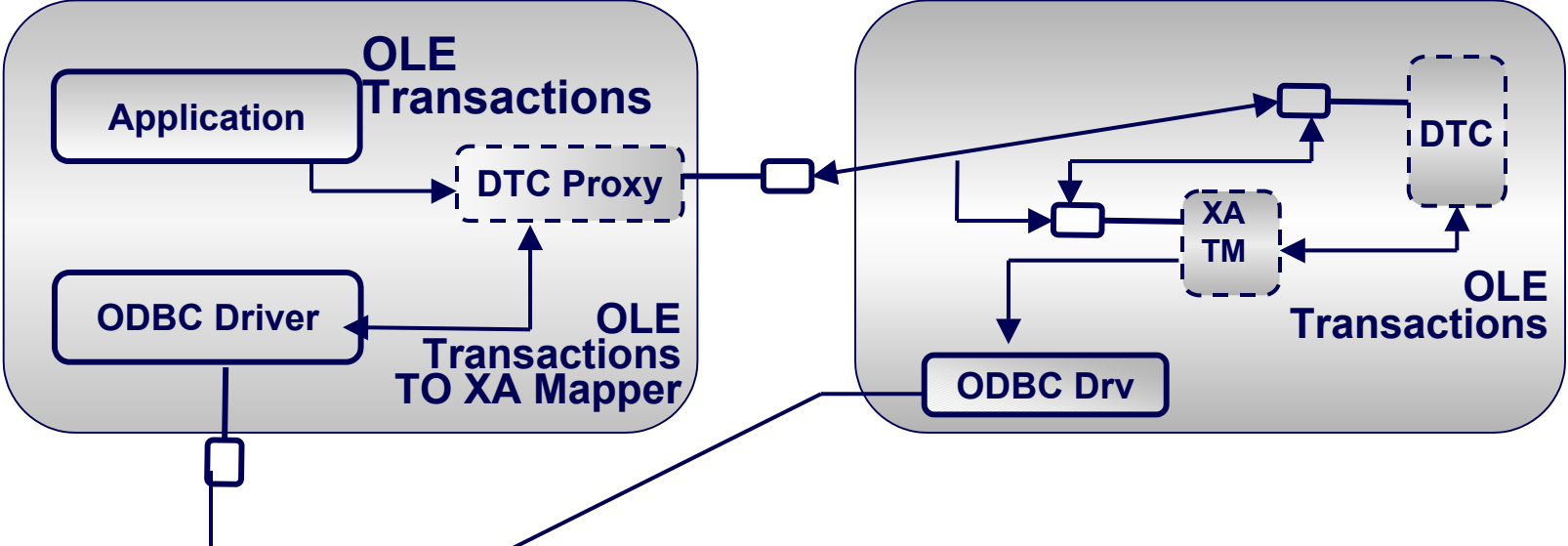
Transaction Processing System

- Three distinct components
 - = Application program
 - = Transaction Manager
 - = Resource Manager



Transaction Processing With XA Compliant Resource Manager

Windows NT Platform



Proprietary API and XA



MS DTC Components - - - - -

Resource Manager Components _____

Any Platform





Two-Phase Commit Protocol

- Applications can either manage transactions themselves (more complex) or use a Transaction Manager
- 2PC is required for distributed transactions.
- 2PC commit procedures are vendor proprietary.
- 2 distinct phases
 - = Prepare
 - = Commit



2-Phase Commit Protocol

- App. chooses Commit agent to co-ordinate transaction
- App. tells RMs to enlist with transaction coordinator
- App. performs actions with RMs
- App. sends PREPARE message to coordinator
- Coordinator sends PREPARE messages to RMs



2-Phase Commit Protocol

- RMs acknowledge Prepare Phase with PREPARED message to coordinator
- If a RM can't prepare or does not respond, coordinator aborts the transaction
- If all RMs are successfully prepared the coordinator tells RMs to commit.



What is MS-DTC?

- Microsoft Distributed Transaction Coordinator
- Creates, destroys, manages, and monitors transactions
- Applications following DTC rules can create Transaction objects and perform operations on them
- Uses OLE Component Object Model



Role of MS-DTC

- Guarantee atomicity of transactions
- Transactions must be initiated by a component of the application
- Component receives a pointer to the transaction
- Pointer cannot be passed
 - = transaction must be completed by initiating component



Application's View of Transactions

- Begin with acquiring a transaction object
- All work is associated with this transaction object
- When in a consistent state it calls a Commit method
 - = If succeeds transaction is committed
 - = If fails the transaction is aborted

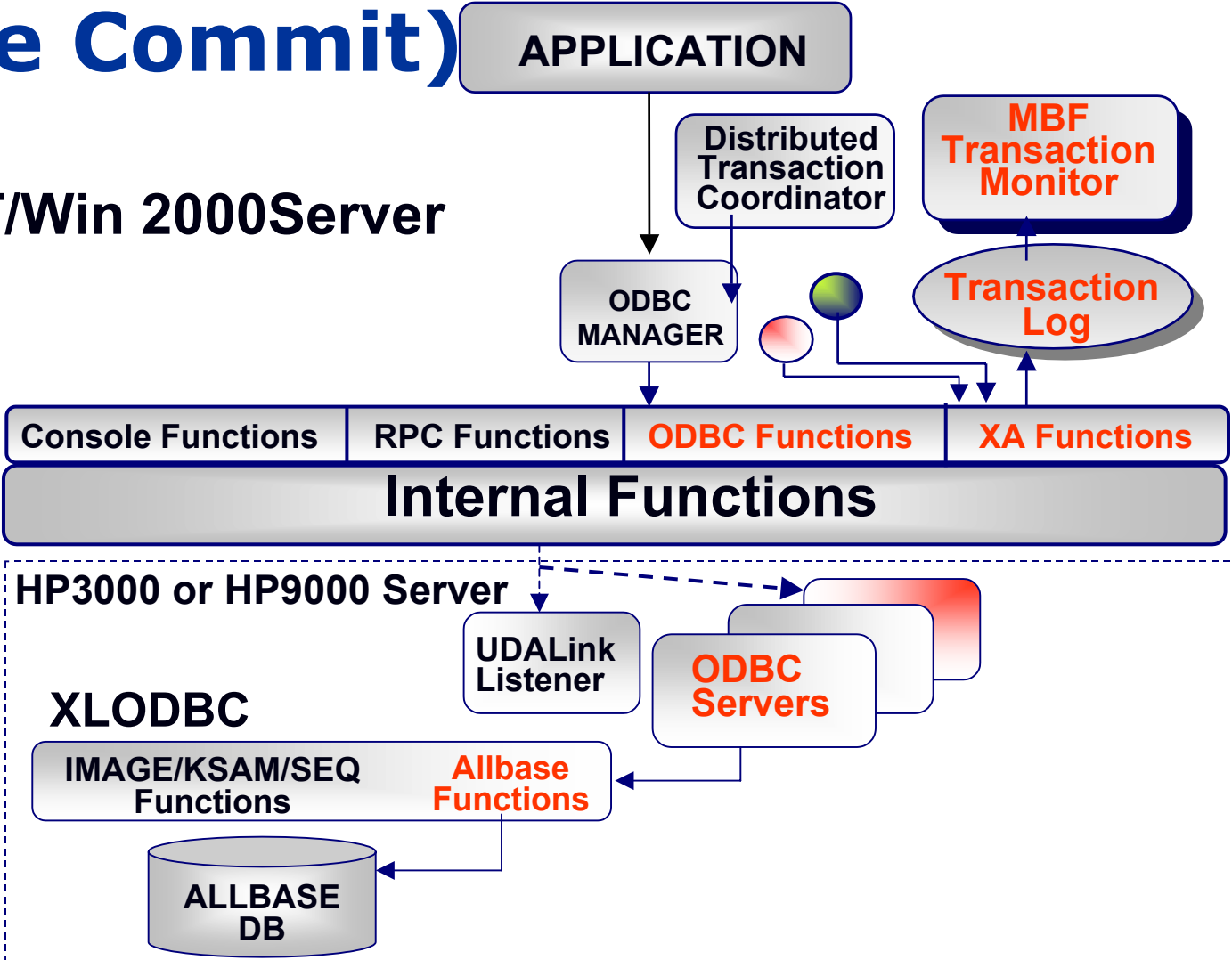


Distributed Transaction

(2 Phase Commit)

WIN-NT/Win 2000Server

UDALink.dll
(Client Driver)





MBF-UDALink & Transactions

- Can ONLY use ALLBASE DBE (NOT Image/SQL)
- No changes to ODBC DSN are required
- Transactions need to be enabled in the Security Catalog



Normal Program Sequence

- Make ODBC Connections
- Initiate Proxy with DTC
- Get transaction handle
- Enlist Connections with transaction
- Perform DB Updates etc.,
- Instruct DTC to Commit Transaction



DTC with COM

- COM object attributes
 - = Does not support
 - = Supports transaction
 - = Requires transaction
 - = Requires NEW transaction
- COM objects hide all transaction code “under the sheets”

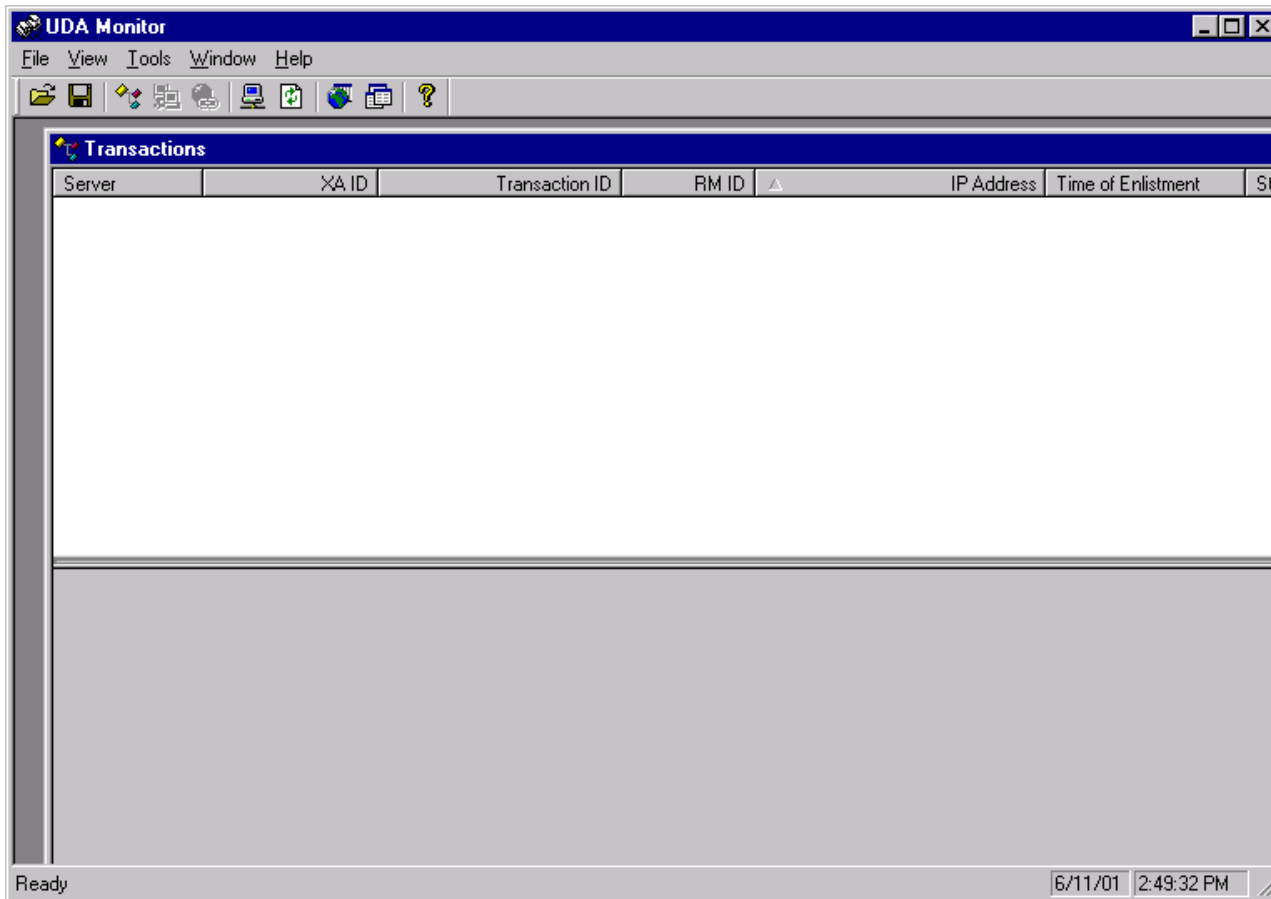


Performance Costs of 2-PC

- If one server is unavailable, no server gets update
- Significant demand for network resources as number of servers increases
- Slower response, as each transaction must wait for ALL parties to complete



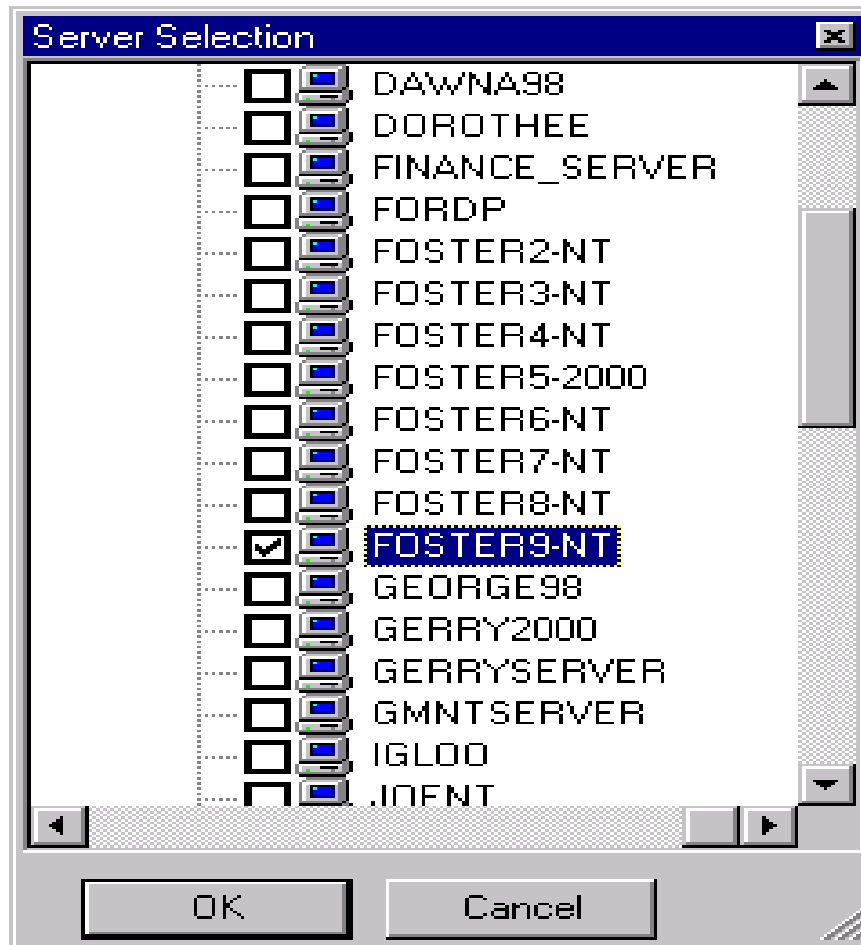
Monitoring Transactions



- Right click to get Server Selection Window



Server Selection Screen



- Select one or more servers to monitor



Transaction Information

The screenshot shows the UDA Monitor application window. The title bar reads "UDA Monitor". The menu bar includes "File", "View", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and help. The main window is titled "Transactions" and displays a table of active transactions. Below the table, there is a summary section for the server "FOSTER9-NT" showing transaction statistics.

Server	XA ID	Transa...	RM ID	IP Address	Time of Enlistment	State	Mode
FOSTER9-NT	58F1458011D55...	17694800	8179312	192.9.3.8	6/11/01 2:51:42 PM	Prepare Requ...	ODBC

FOSTER9-NT:
Transactions Committed: 2314
Transactions Aborted: 0
Transactions Recovered & Committed: 0
Transactions Recovered & Aborted: 0
Time Started: 6/1/01 5:27:58 PM
Last Operation: 6/11/01 2:51:42 PM

Ready 6/11/01 2:51:44 PM

- Top section shows active transactions
- Lower section shows summary statistics

www.mbfoster.com

Forging the Future



Using the RPC Mechanism In the MBF-UDALink Driver

MB Foster



Overview

- What is an RPC call
- MBF-UDALink Architecture
 - = Creating an XL
 - = Creating a COM Component
 - = Creating an Application
 - = RPC Server
 - Parameters
 - Monitoring



What is an RPC Call?

- Execution of a procedure (function) on a different computer.
- Cornerstone of Client/Server development
- Ideally, function appears as if it was on local machine.
- Well established technology
- Many Implementations



COM-TI RPC

- Normally one Component per Server
- Could result in large numbers of components

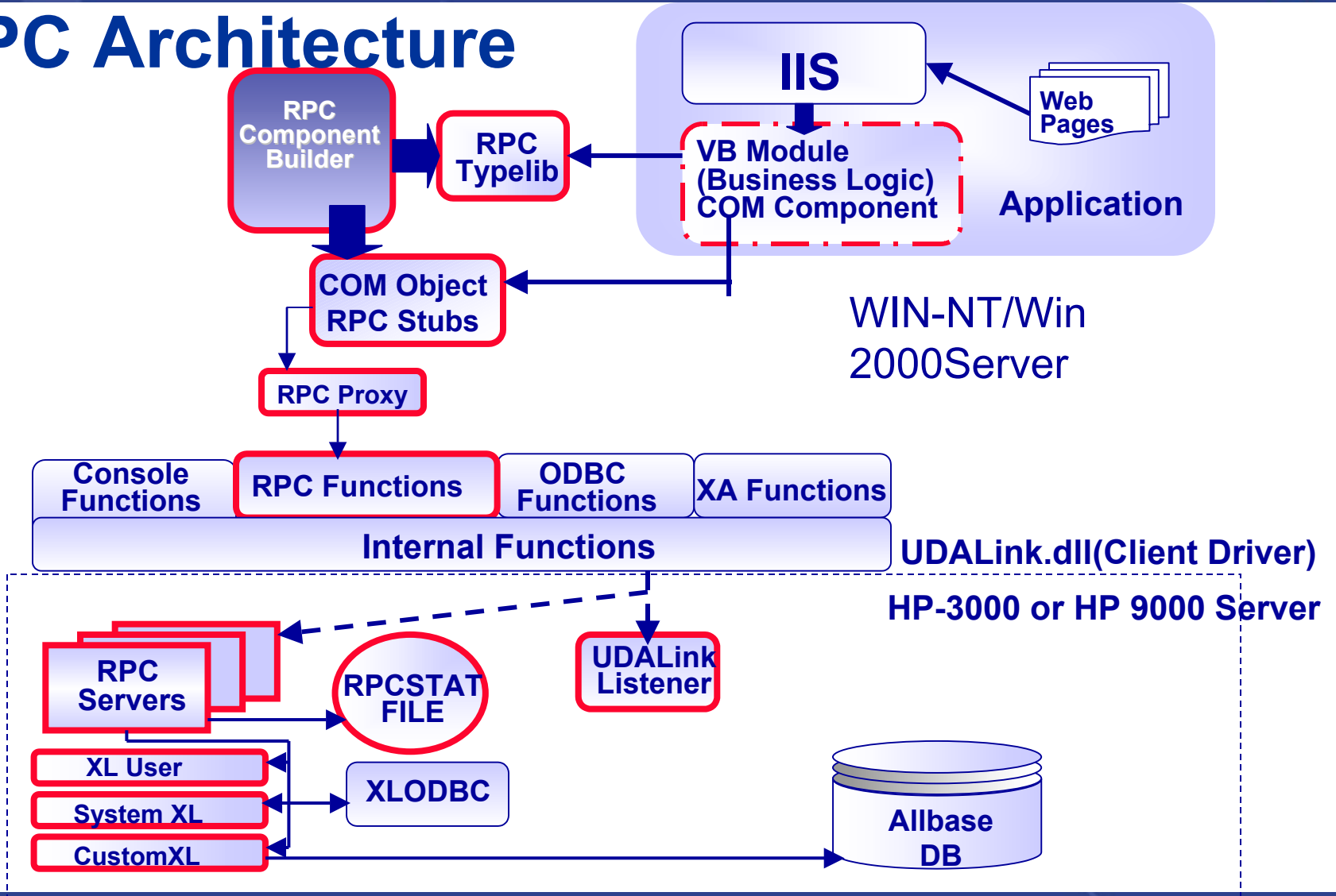


MBF-UDALink RPC

- Many functions are in one COM object
- One COM object may call RPC on ANY server.
- Must establish connection outside the RPC call.



RPC Architecture





Creating an XL

- Almost any procedure that exists in an XL or RL is callable
 - = COBOL, C, PASCAL, FORTRAN
- Datatypes – CHAR, INT, SHORT, BINARY, LONG, REAL, DOUBLE, STRING
- (IN, OUT, or IN/OUT(by reference))
- Structures are allowed



COM Component Builder

- Utility to Generate COM Components which contain the RPC stub
- Supports MFC, MTS, VB Script, Java Script etc.
- Creates TypeLib useable by VB, VC etc.
- Requires Visual Studio 6 to be installed



Using the COM Builder

- Ensure Visual Studio is installed
- Check system Path parameters
 - = C:\Program Files\Microsoft Visual Studio\VC98\Bin;
 - = C:\Program Files\Microsoft Visual Studio\VB98;



COM Builder Output

- Component put in
C:\MBFosterCOM\`<COM_Name>`
- 5 Files created – DONT MOVE !
 - = `COM_Name.IDL`,
 - = `COM_Name.TLB`,
 - = `COM_Name.DLL`,
 - = `COM_Name.LIB`,
 - = and `result.txt`



Using COM Builder - COM definition

- All 3 Names are required

MBF COM Builder

MBF COM Component Builder

Name of New COM :

Name of XL Lib :

Name of RPC Call :

Name of New Structure:

Definition of Parameters

	Type	IN/OUT/REF	Size of Array		Type	IN/OUT/REF	Size of Array
#1:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#11	<input type="text"/>	<input type="text"/>	<input type="text"/>
#2:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#12	<input type="text"/>	<input type="text"/>	<input type="text"/>
#3:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#13	<input type="text"/>	<input type="text"/>	<input type="text"/>
#4:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#14	<input type="text"/>	<input type="text"/>	<input type="text"/>
#5:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#15	<input type="text"/>	<input type="text"/>	<input type="text"/>
#6:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#16	<input type="text"/>	<input type="text"/>	<input type="text"/>
#7:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#17	<input type="text"/>	<input type="text"/>	<input type="text"/>
#8:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#18	<input type="text"/>	<input type="text"/>	<input type="text"/>
#9:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#19	<input type="text"/>	<input type="text"/>	<input type="text"/>
#10:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#20	<input type="text"/>	<input type="text"/>	<input type="text"/>

Parameter/Structure Definition

MB Foster • Forging the Future



- MAX 20 parameters per procedure
- Data Type selected from drop down box
- Defined Structures also show

MBF COM Builder

MBF COM Component Builder

Name of New COM : TestCOM

Name of XL Lib : TestXL

Name of RPC Call : TestCall

Name of New Structure:

Definition of Parameters

Type	IN/OUT/REF	Size of Array	Type	IN/OUT/REF	Size of Array
#1: RPC_CHAR	RPC_IN	20	#11		
#2:			#12		
#3: RPC_CHAR			#13		
#4: RPC_DOUBLE			#14		
#5: RPC_INTEGER			#15		
#6: RPC_LONG			#16		
#7: RPC_REAL			#17		
#8: RPC_SMALLINT			#18		
#9: testa1			#19		
#10: testa2			#20		
#11: A					
#12: ST					
#13:					
#14:					
#15:					
#16:					
#17:					
#18:					
#19:					
#20:					



Adding Methods to Component

- Up to 10000 procedure calls (methods) in one COM object

MBF COM Builder

MBF COM Component Builder

Name of New COM :

Name of New Structure:

Name of XL lib :

Name of RPC Call :

Definition of Parameters

	Type	IN/OUT/REF	Size of Array		Type	IN/OUT/REF	Size of Array
#1:	<input type="text" value="RPC_DOUBLE"/>	<input type="text" value="RPC_IN"/>	<input type="text" value="12"/>	#11:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#2:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#12:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#3:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#13:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#4:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#14:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#5:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#15:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#6:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#16:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#7:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#17:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#8:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#18:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#9:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#19:	<input type="text"/>	<input type="text"/>	<input type="text"/>
#10:	<input type="text"/>	<input type="text"/>	<input type="text"/>	#20:	<input type="text"/>	<input type="text"/>	<input type="text"/>



Data Source Definition

- DSNs are defined using the ODBC Administrator
- ODBC DSNs are useable by RPC
- DSN Name is added as a parameter in the RPC call



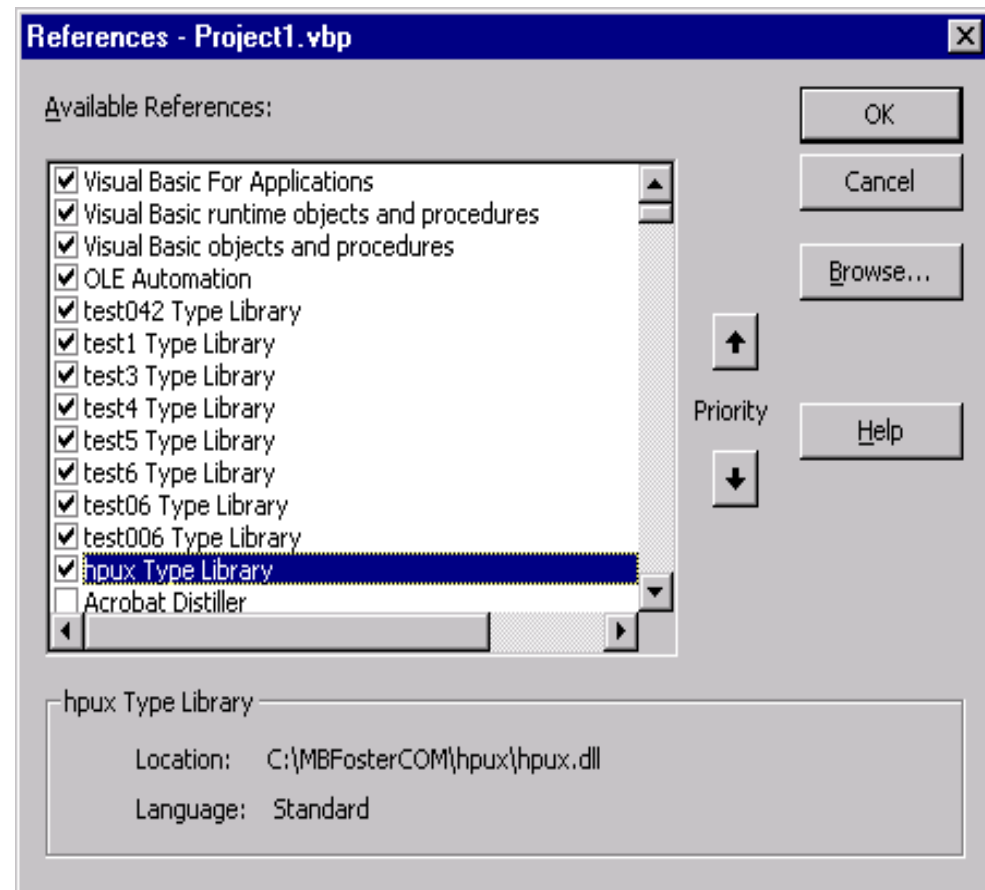
Creating an Application

- C/C++ or VB apps can call the COM Components
- If required the RPC Interface is useable directly (not thru COM)



COM Component with VB

- Select "Project references"
- Check name of the COM Object in "Available References"





RPCConnect Declaration

- Similar to ODBC SQLDriverConnect function
- Public Declare Function RPCConnect Lib "UDALINK3" Alias "_RPCConnect@20" (henv As Long, hdbc As Long, ByVal hWnd As Long, ByVal ConnStr As String, ByVal ConnStrLen As Integer) As Long
- Public Declare Function RPCDisconnect Lib "UDALINK3" Alias "_RPCDisconnect@4" (ByVal hdbc As Long) As Long
- Public henv As Long 'ODBC/RPC environment handle
- Public hdbc As Long 'ODBC/RPC Connection handle



Establishing a Connection

```
Private Sub Form_Load()  
    Dim rc As Long  
    Dim DSNStr As String  
  
    DSNStr = "DSN=RPCTestUX;"  
  
    rc = RPCConnect(henv, hdbc, hWnd, DSNStr,  
        Len(DSNStr))  
    If (rc) Then  
        MsgBox ("RPC Connect failed!")  
    End If  
  
End Sub
```




Simple VB Sample Code

```
Private Sub Command1_Click()  
  
Dim rtn As Long  
Dim ii As Long  
Dim ss As String  
Dim cc As New test042Lib.aaaaa  
  
ss = Text1.Text  
ii = Len(ss)  
cc.Binary hdbc, rtn, ss, ii  
Text2.Text = "rtn = " & rtn  
  
End Sub
```



Internal MBF-UDALink RPC Functions

- RPCConnect
- RPCDisconnect
- RPCInvoke
- RPCSetParam
- RPCGetParam
- RPCNumParams



RPC Server Pool Parameters

- **RPC_INITIAL_SERVERS** – How many processes are initially created in the pool - default is 10. Max is 50
- **RPC_MAXIMUM_SERVERS** – max number of servers that the pool can have. Default 100, Max 100.
- **RPC_MIN_FREE_SERVERS** - min number of free servers allowed. Default 2 Min 1
- **RPC_MAX_FREE_SERVERS** – max number of free servers allowed



RPC Server Monitoring

- Can be seen in the new Console Program
- Can hide unused RPC Server Processes

Console Monitoring

MB Foster • Forging the Future



MB Foster Console File Connect View Window Help

Listeners

Number	Server Name	Port Number	Started at	Version	Jobnum	#ActiveProcesses	#InactiveProcesses	CPU (sec)
1	Fortytwo (M.B.Foster Assoc	21244	2001/06/07 10:02:40	6.02.01	#J152	0	15	0
2	Fortytwo (M.B.Foster Assoc	21266	2001/06/07 10:27:40	6.02.01	#J153	4	0	83.9
3	Fortytwo (M.B.Foster Assoc	21249	2001/06/07 10:29:32	6.02.02	#J154	2	7	1.5
4	Fortytwo (M.B.Foster Assoc	21247	2001/06/07 15:40:44	6.02.02	#J155	4	7	6.4

Show all connections

ODBC-Connections

Number	Server Name	Port Number	Pin	Started at	IP Address	Login	CPU (sec)	Msgs Recvd	Msgs Sent
1	Fortytwo (M.B.Fos	21266	73	2001/06/07 12:05:53	192.9.3.219	INVENTORY	1.8 (0%)	0 (0/per)	0 (0/per)
2	Fortytwo (M.B.Fos	21266	117	2001/06/07 12:06:01	192.9.3.219	ORDERS	2.7 (0%)	31 (3/per)	31 (3/per)
3	Fortytwo (M.B.Fos	21266	122	2001/06/07 12:06:12	192.9.3.219	ACCOUNT	2.1 (0%)	11 (11/per)	11 (11/per)
4	Fortytwo (M.B.Fos	21266	132	2001/06/07 12:06:40	192.9.3.219	ORDERS	1.3 (0%)	0 (0/per)	0 (0/per)
5	Fortytwo (M.B.Fos	21247	74	2001/06/07 15:41:36	192.9.3.138	CONSOLE	1.1 (0.2%)	62 (20/per)	62 (20/per)

RPC-Connections

Number	Server Name	Port Number	Pin	Started at	IP Address	Login	CPU (sec)	Msgs Recvd	Msgs Sent
1	Fortytwo (M.B.Fos	21249	23	2001/06/07 10:29:33	192.9.3.131	MGR.ODBCLI	0.5 (0%)	0 (0/per)	0 (0/per)
2	Fortytwo (M.B.Fos	21249	103	2001/06/07 10:29:34	192.9.3.131	MGR.ODBCLI	0.5 (0%)	0 (0/per)	0 (0/per)
3	Fortytwo (M.B.Fos	21247	133	2001/06/07 15:40:45	192.9.3.138	JOHNM.ODBC	1.9 (3.8%)	13 (11/per)	13 (11/per)
4	Fortytwo (M.B.Fos	21247	60	2001/06/07 15:40:47	192.9.3.138	JOHNM.ODBC	1.2 (2.3%)	3 (3/per)	3 (3/per)
5	Fortytwo (M.B.Fos	21247	124	2001/06/07 15:40:48	192.9.3.138	JOHNM.ODBC	1.1 (0%)	0 (0/per)	0 (0/per)

Connected 6/7/01 3:45 PM

www.mbfoster.com



Forging the Future

Using SSL with MBF-UDALink

MB Foster



Overview

- What is SSL
- Public Key Encryption
- SSL Certificates.
- What is OPEN SSL
- What SSL options
- SSL Limitations



What is SSL

- Secure Sockets Layer
- Industry Standard
- Provides:
 - = Data Encryption
 - = Server Authentication
 - = Message Integrity
 - = Client Authentication (Optional)



Public Key Encryption

- Uses asymmetric keys to encrypt and decrypt
- Public key is distributed
- Private key is secret
- Data encrypted by one key can only be decrypted by the other



Handing out Public Keys

- Can be compromised if this is all there was
- You can say you are someone you are not then hand out a public key for others to use
- A 'certificate' is needed



SSL Certificate

- Certificate contains:
 - = Issuers name
 - = Entity (subject) that certificate is issued to
 - = Public Key of subject
 - = Some time stamps
- Certificate BINDs a public Key to a name

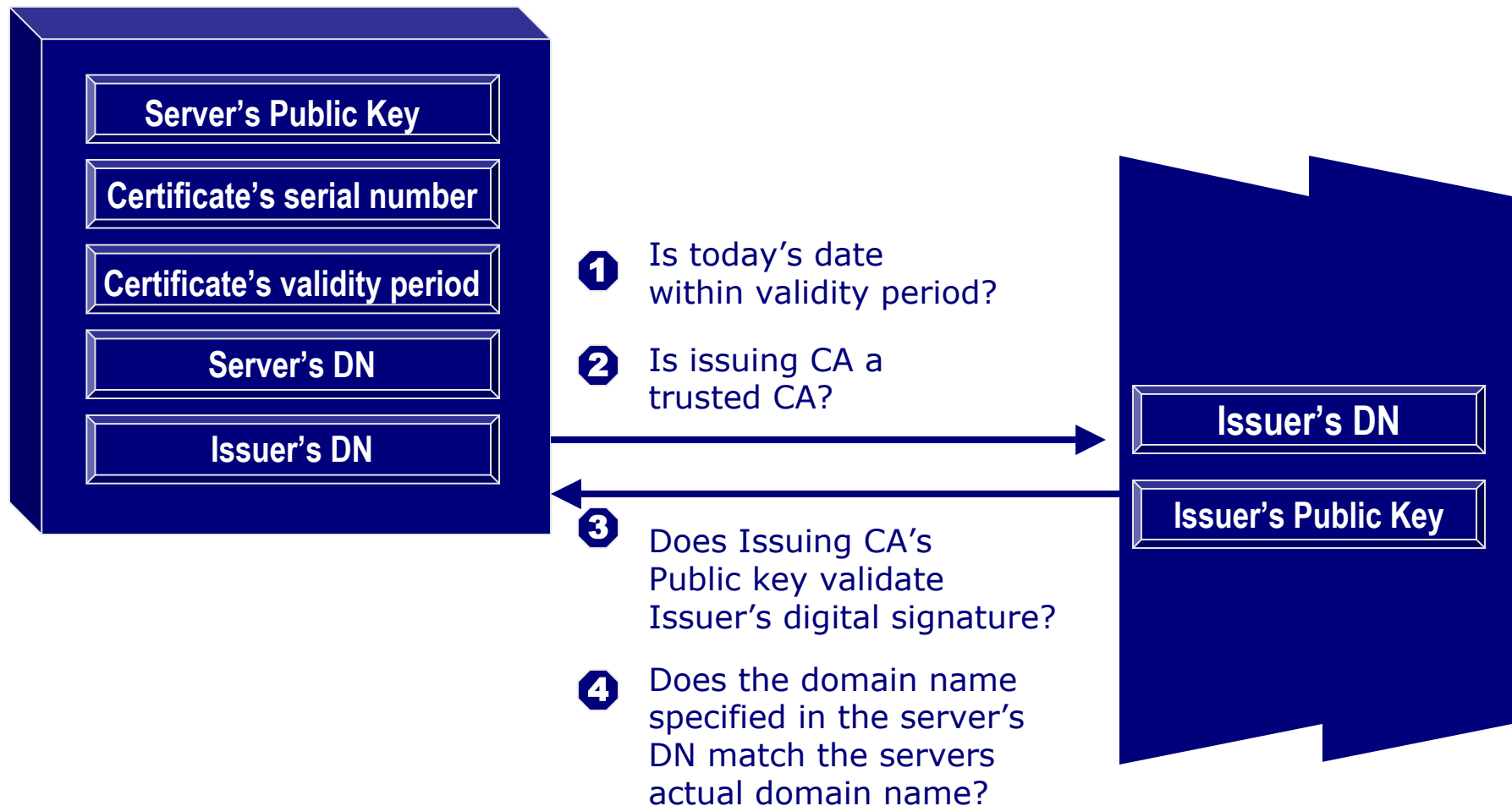


Server Certificates

- Server certificates are designed to protect those connecting to the Server.
- Allows Server Authentication
- Keeps Private Communications Private



Server Certificate Authentication





Client (Personal) Certificates

- Restrict Access to Specific Consumers
- Authenticate a Client
- Can simplify access
- Allows Server to keep track of Clients



Certificate Generation

- Normally done by certificate Agent
 - Entrust
 - VeriSign
 - SecureNet
 - CyberTrust
 - CertiSign
 - etc.
- You can use Open SSL to generate your own certificate pairs.



What is OPEN SSL

- Cryptography toolkit
- Implements SSL V2/V3 and TLS v1
- Has a command line tool for numerous crypto functions
- Can create Keys, digests, and Certificates



SSL Implementation

- SSL Setup is done in Client setup
- Must use UDALink3S.dll
- Server SSL Code is in a separate XL- XLSSL



SSL Options

- supports the following encryption schemes:
 - = TLS 1 (Transport Layer Security)
 - = SSL V2
 - = SSL V3
- supports two Key Exchange modes
 - = RSA
 - = Diffie Hellman (DH)



SSL Client Setup

- Simply check the SSL Box and click OK

MB Foster UDALink

Change data source description, connection type, and options.
Then choose OK.

Data Source Name:

Description:

Direct serial Modem Winsock

© 1995-2001 M.B. Foster Associates

Host Configuration Login

HostConf User-ID ('?' will prompt) Hostconf Pwd ('?' will prompt)

Host Server Info

Server name or IP Address Socket ID (Port on host) (Leave blank to use default)

ChangeLogon Use SSL

Max Stmt's per Connection (1-48)

Timeouts

Command Sort

Debug Logging

Client Logging

Host Logging

OK

Cancel



SSL Configuration

SSL Configuration [X]

SSL Version

- SP_PROT_TLS1
- SP_PROT_SSL2
- SP_PROT_SSL3

Exchange Key Mode

- CALG_DH_EPHEM
- CALG_RSA_KEYX

Certificate Server Common Name:

Server Certificate Authorization

Cancel OK



Server SSL Configuration

- The following Environment variables must be set.
 - = MBF_SSL_ENABLE = 1
 - = MBF_SSL_VERSION = ? (1, 2 3 or 4)
 - 1 = SSL v2
 - 2 = SSLv3
 - 3 = dynamic negotiation of SSL v2 or v3
 - 4 = TLS V1
 - = MBF_SSL_CERTIFICATE_FILE= server cert location
 - = MBF_SSL_KEYFILE = private key location



Special Notes

- Adding UseSSL=1; to the connect string without having Server Certificate information will simply use a standard encrypt/decrypt hashing algorithm without doing certificate verification



SSL Limitations

- does not currently support Client Certificate Authentication



Contact Us

By Phone: *1-800-ANSWERS (267-9377)*
613-448-2333

By Fax: *613-448-2588*

By E-Mail: *Support@mbfoster.com*

On the Web: *www.mbfoster.com*