Central Bank Foreign Exchange

Table of Contents

Introduction

- This is an introduction of Central Bank Foreign Exchange solution pack by Cedar Software.
- The main focus of this document is to help customer evaluate the BlueStar solution.

Advantages of CEDAR Solution on Central Bank Foreign Exchange

The primary advantages of CEDAR solutions are listed below.

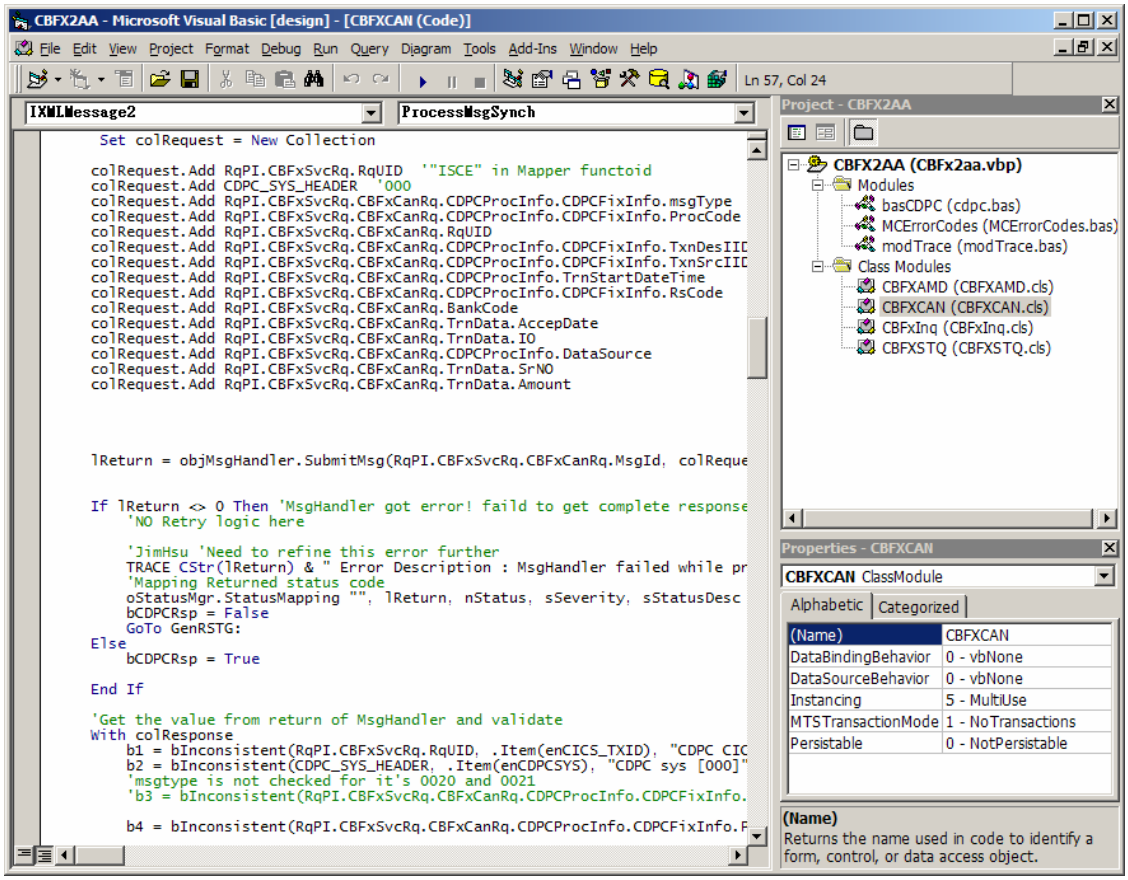
Modularity and component help separate the technologies and business rules, provide ease of maintenance

The solution is based on component and XML. There is clear separation between transport, data type parsing, encoding translation, data transformation and business rules. Every specific task is resolved in one

specific component, so that revising and extending the function in one module will not affect the other part of the solution.

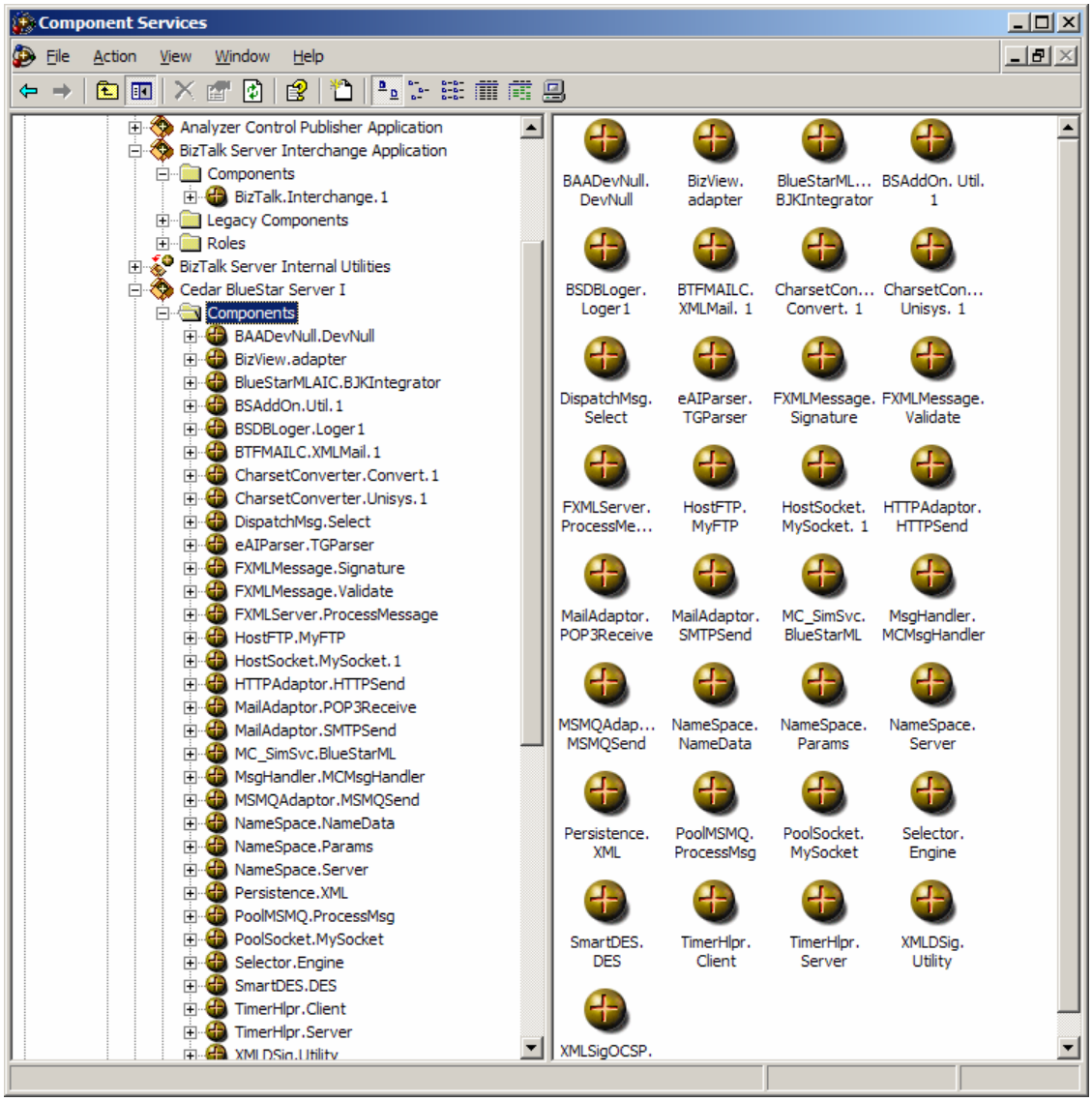
When customers adopt the solution, they will notice that there are quite a few components and layers in our solution, they are an achievement of careful componentization in the solution. And normally only the "business rule" components in them need to be revised when business rules change.

For example, the Transport (LU0, LU6.2) component, the data type (X, 9, Comp-1, Comp-3, Native…), the Encoding (Unicode, ASCII, EBCDIC, Unisys) and the complex file layout (delimited, positional, variable positional). In most other monatomic solutions, these complex issues are not resolved separately, a slightly revision for business rule in the solution will require a full understand of all the above. This will not happen with our solution, since all these are separated. Most of the time, a revision with business rule will only need the business rule component been touched.



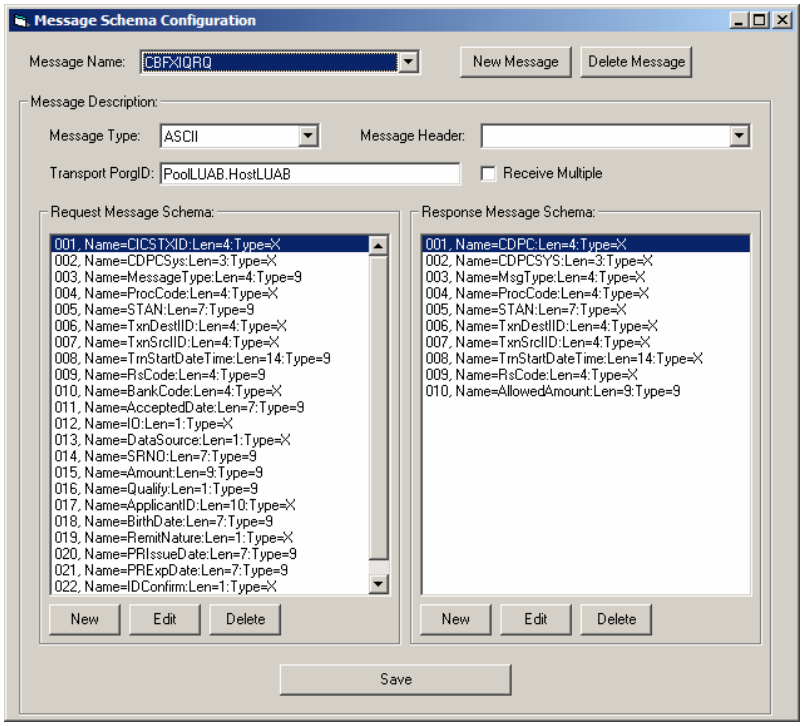
Single binary shared modules and components imply robustness and solid performance

All these components are single binaries shared among a rich list of customers in productional environments. The robustness and performance are rock solid and well-proved.



Design and development in Graphic User Interface, least code for business rules needed

Most of the component are glued together in this solution by GUI Configuration, this again, makes lease code revision needed when there is change in the spec, for example, if a new column to host is needed, the schema of the host telegram, the mapping are all configured in the GUI.



In addition to the significant advantage of TCO, the solution utilizes the COM+ and .NET solution which provide best performance and openness.

COM+ Connection pooling

The bottleneck of the transport is coded to support connection pooling, which will make the request goes through a "host session". This will save the application from building up session with host every time new requests need to be sent and significantly boost the performance.

XML Business Schema

Since XML is positioned at the heart of this solution, the telegram is transformed to XML data in the internally. This openness allows future extension of channels to access it though this XML Schema.
There is a complete and comprehensive logging now, and it's pretty easy to extend from the XML business Schema.
The objective of multi-channel integration is at the heart of this design.

XML Web Service readiness

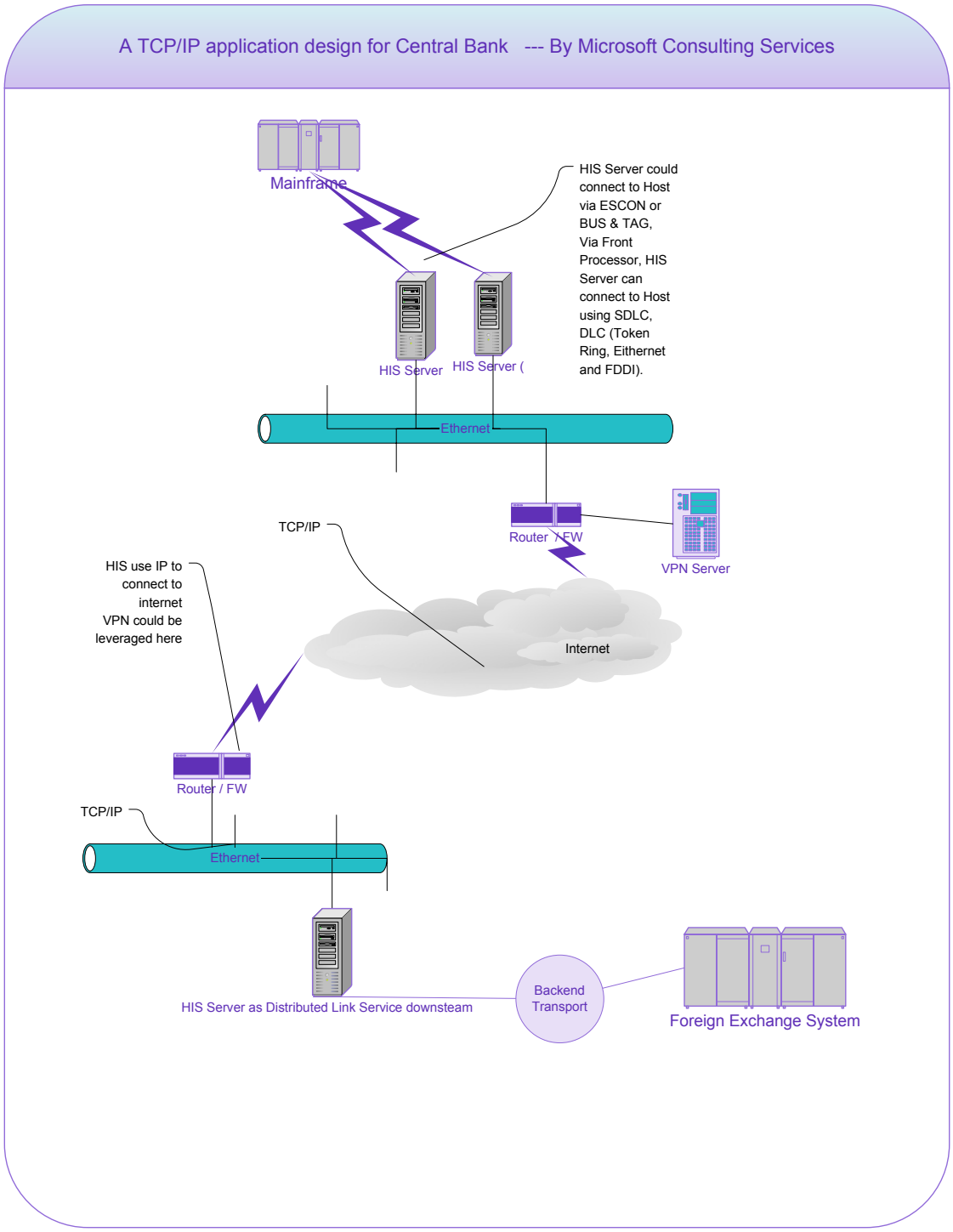
The solution is built on XML Openness, so it's ready to be exported to XML WEB Services.

Requirements

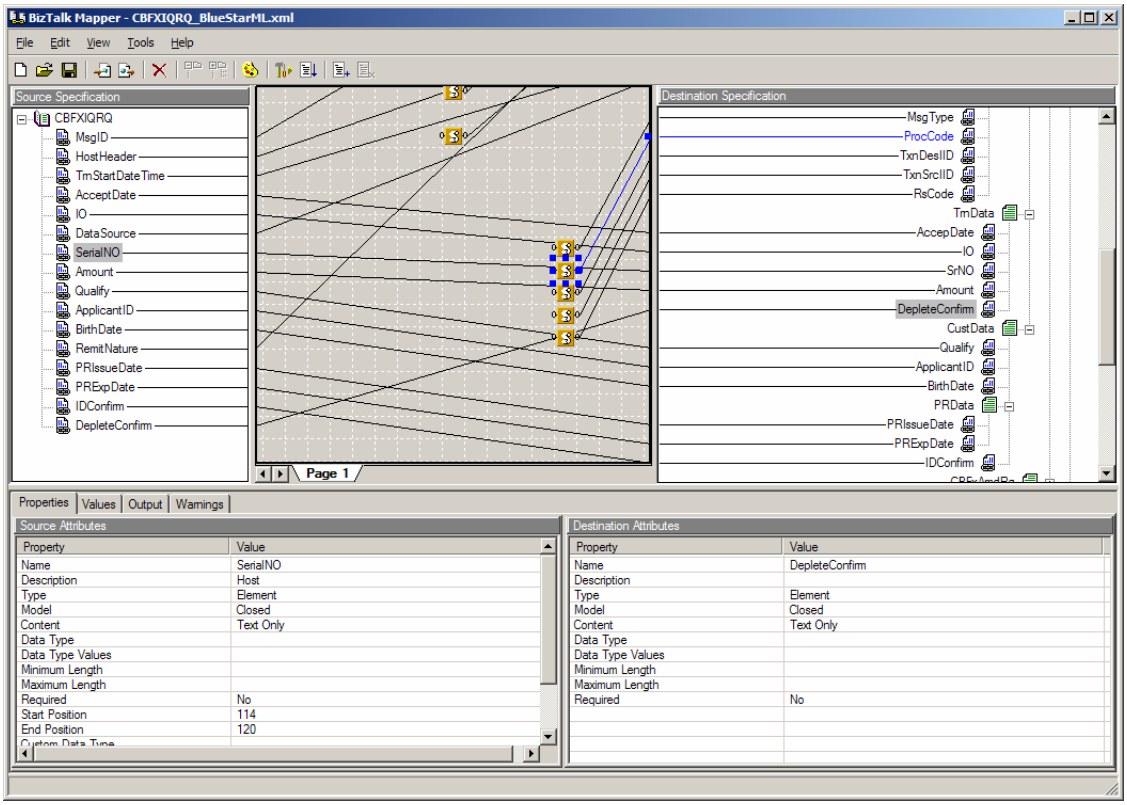
The POC needs to connect to CUSTOMER Host system and receive APPC request from Host. Also demo the CDPC connection.

Architectures

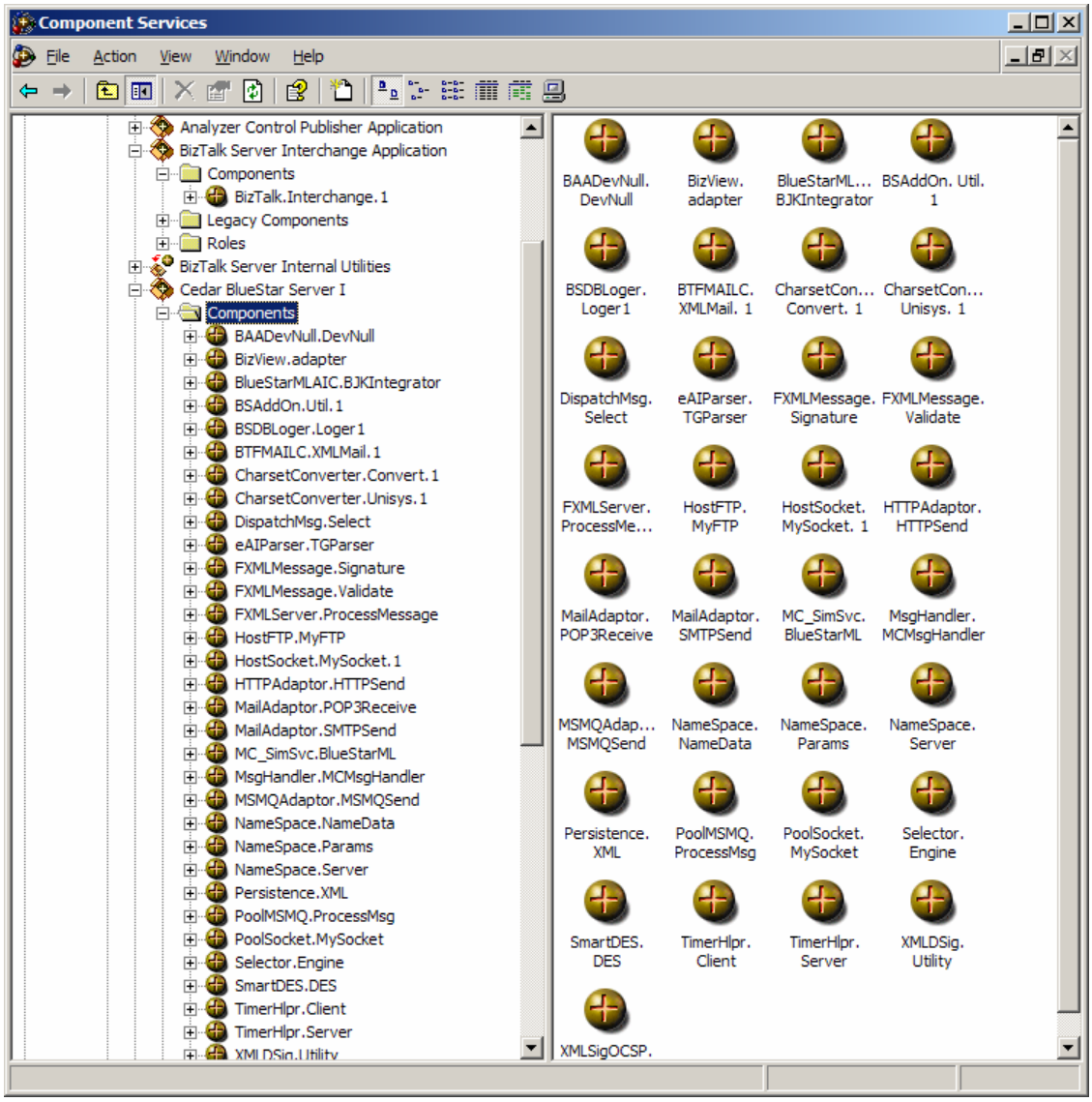
On the Application Server:
Microsoft HIS server is needed to connect to CDPC via LUA over DLS. Also the same HIS server will be configured to connected, Receive and Send APPC call from and to HOST of CUSTOMER
Microsoft BizTalk Server is used for the XML Schema, Mapping and Validation.
A limited license of BlueStar Server is used for the application platform and Host Telegram Translation. This is the key architecture of our solution.
Microsoft SQL Server is used for the configuration, message Schema and Log storage. BizTalk Server also needs SQL Server.



The Host telegram from the Foreign Exchange system is transformed into business schema in XML



The application login is built into COM+ components,



We use COM+ Connection pooling in LUA transport, which provide excellent performance in this solution.

Preparation work for CUSTOMER for customization

1. CUSTOMER need to prepare 1 port of Ethernet internet connection (by pass firewall if possible, since DLS will now need dynamic port) the IP address. The DNS.
2. Also 1 connection ready for Back end Foreign Exchange system, notice this 2 ports will be used to connect to the same testing PC Server
3. A working seat and maybe a temporary card key for CEDAR to work in CUSTOMER.
4. CUSTOMER will need to define the node of this from Host
5. CUSTOMER will need to prepare the host side program.

Message lists

The Message of testing is from the list that CDPC provided below, however, now that CDPC won't send Connection Request, and CDPC does not need us to send connection request either, so the 1.2.3 message is no longer needed any more. What has been tested is 4 – 11.

Ref. No.	Transaction	Transaction ID	Transport Protocol	使用單位	
				Sender	Receiver
1.	CBC Connection Request	CBOutConnRq	TN-3270 or DLS	Multi-Channel	CDPC
2.	CBC Inbound Connect Request	CBInConnRq	TN-3270 or DLS	Multi-Channel	CDPC
3.	CBC Inbound Connection Response	CBInConnRs	TN-3270 or DLS	CDPC	Multi-Channel
4.	Central Bank Foreign Exchange Inquiry Request	CBFxInqRq	TN-3270 or DLS	CDPC	Multi-Channel
5.	Central Bank Foreign Exchange Inquiry Response	CBFXInqRs	TN-3270 or DLS	Multi-Channel	CDPC
6.	Central Bank Foreign Exchange Amendment Request	CBFxAmdRq	TN-3270 or DLS	CDPC	Multi-Channel
7.	Central Bank Foreign Exchange Amendment Response	CBFxAmdRs	TN-3270 or DLS	Multi-Channel	CDPC
8.	Central Bank Foreign Exchange Inquiry Cancellation	CBFxCanRq	TN-3270 or DLS	CDPC	Multi-Channel
9.	Central Bank Foreign Exchange Cancellation Response	CBFxCanRs	TN-3270 or DLS	Multi-Channel	CDPC
10.	Central Bank Foreign Exchange Status Inquiry Request	CBFxStatusInq Rq	TN-3270 or DLS	CDPC	Multi-Channel
11.	Central Bank Foreign Exchange Status Inquiry Response	CBFxStatusInq Rs	TN-3270 or DLS	Multi-Channel	CDPC

Sample messages

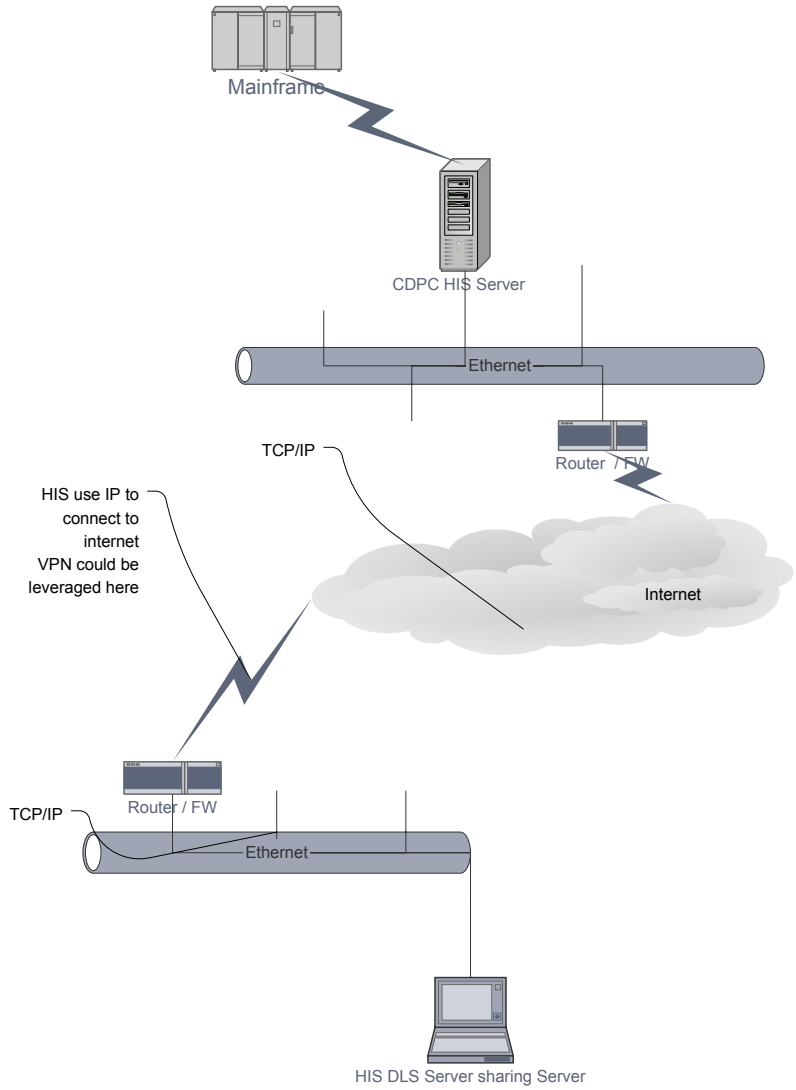
See attachment.

Testing environments

Network topology

Testing is done from testing HIS 2000 Server, and the sending client.

Testing Environment



The testing request has been issued from requesting bank, this is an HIS server running, it is sharing a Link Service shared by CDPC HIS Server. The request travels from requesting bank via Intranet, to CDPC HIS Server via Intranet.

Testing program

Testing is done on a testing utility developed by CEDAR; all data and schema are in XML and XML Schema, testing data could be modified and saved easily so that repeated testing could be flexible. Below are the snapshots of this testing utility