

Fujitsu Services **High Level Design Specification for Agents for NBX,
the NBE Replacement** **Ref: NB/HLD/017**
COMPANY-IN-CONFIDENCE **Version: 2.0**
 Date: 13/03/2006

Document Title: **High Level Design Specification for Agents for NBX, the NBE Replacement**

Document Type: High Level Design

Release: BI3 S75 and beyond

Abstract: This document is the High Level Design (HLD) for the new and changed Agents produced for NBX, the NBE Replacement. It is an internal Fujitsu Services document. The level of detail is intended to act as a baseline to Fujitsu Services (Post Office Account) developers and testers.

Note that this is a separate HLD from that for the former Network Banking System (NBS) Agents that interfaced with the external Networking Banking Engine (NBE). Both sets of Agents will have a short period of co-existence.

Document Status: APPROVED

Originator & Dept: Rex Dixon (Tel: GRO) / Development Unit - Design Team

Contributors: –

Internal Distribution: POA Document Management, Agent Document Library

External Distribution: None

Approval Authorities:

Name	Position	Signature	Date
Tom Northcott	SI Design Manager		
Roy Birkinshaw	SI Development Manager		

0. Document Control

0.1 Document History

Version No.	Date	Reason for Issue	Associated CP/Pin/ICL No.
0.1	16/09/2004	First draft	
1.0	26/11/2004	Version for approval	
1.1	23/12/2005	Changes and corrections made during implementation. Addition of performance information previously missing	CP3896 PC0111214 PC0112176 PC0112451
2.0	13/03/2006	Version for approval, incorporating a few minor changes as a result of the formal review	

0.2 Review Details

Review Comments by :	
Review Comments to :	Originator

Mandatory Review Authority	Name
ASS Designer	Allan Hodgkinson(*)
SI Development Manager	Roy Birkinshaw(*)
CS System Support Centre Manager	Mik Peach(*)
CS Network Service Manager	Alex Kemp(*)
CS Data Centre & Ops Service Manager	Peter Thompson
CS Security Manager	Brian Pinder(*)

Optional Review / Issued for Information	
SI Team Leader	Peter Ambrose
SI Designers	Simon Fawkes; James Stinchcombe; Sudhanshu Agrawal; Tom Northcott; Alex Robinson(*)
SI Test Designer	Peter Robinson
CS Infrastructure & Availability Manager	Carl Marx
CS Service Introduction Manager	Graham Welsh
CS Service Definition Manager	Jane Collins
CS Release Manager	John Budworth
SI Development Team	Anne Mohan(*); John Rayner; Mike Conneely
ASS Technical Designer	Mark Jarosz; Simon Fawkes; David Chapman

(*) = Reviewers that returned comments

0.3 Associated Documents

Reference	Document	Vers.	Title	Source
[A&L_AIS]	NB/IFS/026		NBX – A&L Application Interface Specification (AIS)	PVCS
[A&L_MAP]	NB/IFS/034		Horizon – A&L Mapping	PVCS
[A&L_TIS]	NB/IFS/029		NBX – A&L Technical Interface Specification (TIS)	PVCS
[ACKFS]	AD/FSP/001 (TSC/AGT/065)		Functional Specification for a Generic Acknowledgement Agent for CSR+	PVCS/ agent lib
[EACRRSOD]	SY/SOD/002		ACRR Improvements for Network Banking – System Outline Design	PVCS
[BUSPARAMS]	NB/IFS/035		NBX Business Parameters	PVCS
[BUSVOLS]	PA/PER/033		Horizon Capacity Management and Business Volumes	PVCS
[CAPO_AIS]	NB/IFS/025		NBX – CAPO Application Interface Specification (AIS)	PVCS
[CAPO_MAP]	NB/IFS/031		Horizon – Card Account Mapping	PVCS
[CAPO_TIS]	NB/IFS/027		NBX – CAPO Technical Interface Specification (TIS)	PVCS
[CAVEATS]	CR/REP/051	1.0	NBE Replacement - Conceptual Design Contractual Caveats and Exclusions	PVCS
[CD_NBX]	BD/CDE/005	4.0	NBE Replacement – Conceptual Design	PVCS
[COMAGT]	AD/DES/042 (TSC/AGT/076)		High Level Design of Common Agents	PVCS/ agent lib
[CRYAPI]	RS/DES/024		Cryptographic Functions API	PVCS
[CRYPTOAPI]	RS/IFS/001		Cryptographic Application Programming Interface Specification	PVCS
[DP_NBX]	AS/DPR/009	2.0	Design proposal for EMV, TDES and NBE Replacement	PVCS
[DRSC12AIS]	NB/IFS/007		Product Interface Specification: [C12] Confirmation Agent – DRS	PVCS
[DRSHLD]	NB/HLD/003		NWB Data Reconciliation Service High Level Design	PVCS
[EMVCTR]	NB/HLD/012		EMV Retail and Banking, Counter High-Level Design Specification	PVCS
[ETSHLD]	AD/DES/073		High Level Design Specification for the Electronic Top-Up Agents	PVCS/ agent lib
[GENAGT]	AD/DES/039 (TSC/AGT/058)		Generic Agent Components for CSR+ High Level Design	PVCS/ agent lib

**Fujitsu Services High Level Design Specification for Agents for NBX,
the NBE Replacement
COMPANY-IN-CONFIDENCE**

**Ref: NB/HLD/017
Version: 2.0
Date: 13/03/2006**

[GENTABS]	AD/SPE/006 (TSC/AGT/079)		Agents Generic Database Table Specifications for BI3	PVCS/ agent lib
[HADDIS]	TD/STD/001		Host Applications Database Design and Interface Standards	PVCS
[KMSHLD]	RS/DES/010		Key Management High Level Design	PVCS
[LINK_AIS]	NB/IFS/024		NBX – LINK Application Interface Specification (AIS)	PVCS
[LINK_MAP]	NB/IFS/033		Horizon – LINK Mapping	PVCS
[LINK_TIS]	NB/IFS/028		NBX – LINK Technical Interface Specification (TIS)	PVCS
[LUC]	AD/DES/036 (TSC/AGT/081)		Cluster Lookup Service Design	PVCS/ agent lib
[MAESTRO]	AD/DES/032 (TSC/AGT/063)		Design of Agent Maestro Schedules for CSR+	PVCS/ agent lib
[MIGRATION]	NB/STR/015		S70/S75 Migration Strategy	PVCS
[MSGFLOWS]	NB/IFS/004		Network Banking Message Flows and Interfaces	PVCS
[NBCONF]	AD/LLD/001 (TSC/AGC/029)	3.0	Detailed Design for the NBS Confirmation Harvester Agent	PVCS/ agent lib
[NBMON]	SY/DES/011 (SMG/DES/0001)		Network Banking Monitoring – Core Functionality	PVCS/ SMG
[NBSHLD]	AD/DES/065		High Level Design Specification for Network Banking Agents	PVCS/ agent lib
[NBXAUTH]	NB/LLD/063		Design Specification for the NBX Authorisation Agents	PVCS/ agent lib
[NBXCRYAPI]	RS/LLD/013		NBX Crypto API Low Level Design	PVCS
[NBXGREV]	AD/LLD/005		Low Level Design for NBX Guaranteed Reversals Agent	PVCS/ agent lib
[NBXJNL]	NB/IFS/040		Specification of NBX Journal Records	PVCS
[NBXMON]	SY/DES/032		NBX Online service reporting and alerting specification	PVCS
[NBXNETWORK]	NB/HLD/024		NBX Network Infrastructure High level Design	PVCS
[NBXROUTING]	NB/LLD/064		NBX Routing Agent Low Level Design	PVCS/ agent lib
[NPS]	NB/HLD/013		NBX Persistent Store HLD	PVCS
[OMDB]	AD/DES/062 (TSC/AGT/084)		OMDB Agents High Level Design	PVCS/ agent lib
[OPOVER]	AD/PDN/001 (TSC/AGT/072)		Pathway Agents: Operational Overview	PVCS/ agent lib
[PERFMON]	AD/DES/037 (TSC/AGT/083)		Agent Performance Monitoring Libraries Design for CSR+	PVCS/ agent lib

[PPDAUTH]	NB/DES/007		Platform Physical Design for the NBX Authorisation Agent Server	PVCS
[PPDROUTING]	NB/DES/008		Platform Physical Design for the NBX Routing Agent	PVCS
[SECBUILD]	RS/DES/081		Implementation Build Guide for Secure NT Platforms	PVCS
[SECFS]	RS/FSP/001		Security Functional Specification	PVCS
[SECPOL]	RS/POL/002		Horizon Security Policy	PVCS
[SECROLES]	RS/REQ/022		Secure Role Definitions for SECURENT Build	PVCS
[TED]	TD/ARC/001		Technical Environment Description	PVCS
[TEM]	PA/TEM/013		High Level Design Specification template	PVCS
[TESELEM]	NB/DES/006		Transaction Enquiry Service (TES) Elements Specification	PVCS
[TESPOREPORTS]	NB/IFS/036		Transaction Enquiry Service (TES) Post Office Reports Specification	PVCS
[TPSHLD]	AD/DES/041 (TSC/AGT/074)		TPS Agents for BI3: High Level Design	PVCS/ agent lib
[TPSTABS]	AD/DES/047 (TSC/AGT/075)		Horizon Agents: TPS Tables and Mappings for BI3 and beyond	PVCS/ agent lib

Where explicit versions are specified, these are the versions that have been consulted in the preparation of the current document.

0.4 Abbreviations/Definitions

0.4.1 Abbreviations

[A]	Authorisation returned from the FI to the Horizon Counter
[A1]	Authorisation message returned from the FI to the NBX Authorisation Agent
[A3]	Authorisation message returned from the NBX Authorisation Agent to the Horizon Counter, normally a transformation of an [A1]
[A4]	Variant of an [A3] that is not returned to the Counter, generated when signing the [A3] failed. It is for diagnostic purposes only
ACT	Agent Checkpoint Table
AIS	Application Interface Specification; standard document type required for each interface to the Horizon system
A&L	Alliance & Leicester
API	Application Programming Interface
ART	Agent Run-State Table
ASTS	Authorisation Agent Status message in the inter-Agent protocol
AWK	Acquirer Working Key, the key used to encrypt the PIN Data in the [R3] sent to the FI_EE

AZMK	Acquirer Zone Master Key
[C]	Confirmation message
[C0]	Message sent from the Counter indicating that a potentially Authorised transaction did not complete, or that no Authorisation was received within the time-out period
[C1]	Confirmation message written as an NBS EPOSS Transaction
[C12]	Transformation of a [C1] as written to the DRS
CAPO	Post Office Card Account
CS	[POA's] Customer Services organisation
CSR+	Core Service Release plus
DCS	Debit Card System
DLL	Dynamic Linked Library
DP	Design Proposal
DRS	Data Reconciliation Service
DWh	Data Warehouse
[E1]	Reversal message sent to an FI
[E2]	Reversal confirmation message received in response to an [E1]
EACRR	Enhanced Agent and Correspondence Server Resilience & Recovery
EoD	End of Day
EPOSS	Electronic Point of Sale Service; Horizon service that supports retail functions in Outlets
ETS	Electronic Top-Up Service
FAD	Finance Accounts Division; part of PO Ltd
FI	Financial Institution, one of CAPO, LINK or A&L
FI_EE	FI Enquiry Engine, the collection of those of the FI's PIs corresponding to a Logical FI (e.g. CAPO_A)
FTMS	File Transfer Management Service; Horizon process that provides configurable file transfer services between Horizon and POL's Clients. Services available include data compression and encryption
FTP	File Transfer Protocol
HSM	Hardware Security Module
IP	Internet Protocol
KMA	Key Management Application
KMS	Key Management Service
LAN	Local Area Network
LST	Live System Test
LUC	Cluster Lookup Service
MAC	Message Authentication Code
ms	millisecond(s)
MTBF	Mean Time Between Failures
NB	Network Banking

NBA	Network Banking Application
NBE	Network Banking Engine; system that handles the interface between the Horizon system and the <i>Financial Institutions</i> (FIs) that have reached agreement to provide automated banking services in Post Office Outlets
NBS	Network Banking Service; the acronym used in this Document for the application that supports banking functionality within the Horizon architecture
NBX	NBE Replacement
NPS	NBX Persistent Store
OLA	Operational Level Agreement
OMDB	Operational Management Database
PAN	Primary Account Number
PI	Process Interface
PIN	Personal Identification Number
PKC	Public Key Certificate
POA	Post Office Account (team)
POL	Post Office Ltd
PPD	Platform Physical Design
QOS	Quality of Service
[R]	Request sent from the Horizon Counter to an FI
[R1]	Request message sent from the Horizon Counter to the NBS Authorisation Agent
[R3]	Transformation of an [R1] as sent to an FI
RAC	Request/Authorise/Confirm
RAG	Riposte Attribute Grammar
RDDS	Reference Data Distribution System
RDMC	Reference Data Management Centre
RDMS	Reference Data Management Service
RDS	Reference Data System; POL system that provides a Reference Data feed to Horizon and other systems
RDT	Reference Data Team [within POA]
RPC	Remote Procedure Call
RSTS	Routing Agent Status message in the inter-Agent protocol
SLA	Service Level Agreement
SMDB	Service Management DataBase
SQL	Standard Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
TIP	(POL's) Transaction Information Processing system
TIS	Technical Interface Specification; standard document required to specify and agree the technical details of each external interface
TMS	Transaction Management System

TPS	Transaction Processing Service; Horizon service that formats data for transmission to TIP
UTC	Stands for Co-ordinated Universal Time (ISO 8601); equivalent to Greenwich Mean Time (GMT), but a term that doesn't upset some of our continental neighbours
XML	eXtended Mark-up Language

0.4.2 Definitions

The following terms, when Capitalised as here, have specific meanings as indicated.

Active Agent	Of a Resilient Agent Pair, the instance that is actively processing the workload
Agent	Component of the Horizon Application Architecture that conventionally sits between the Correspondence Servers and Host (or external) layers
Agent Server	Hardware platform that supports Agent processes. It includes the <i>Generic Agent Server</i> , as well as application-specific servers such as the <i>NBX Authorisation Agent Servers</i> and <i>NBX Routing Agent Servers</i>
Approval	The FI verdict contained in an <i>Authorisation</i> [A] that permits an NBS transaction to complete according to the Request [R]. The Clerk or Customer may still prevent the transaction from completing e.g. bad signature or change of mind
Audit Trail	One or more Audit Tracks, which between them, enable an auditor to follow the treatment of related data transfers, movements or accesses by named individuals
Authorisation Agent	Agent that processes [R] messages from Counters, and passes them to the NBE, then taking [A] messages from the NBE and passing them to the Counter
Branch	Post Office location with one or more Counter PCs installed as part of the Horizon programme
Bulk Agent	Agent which reads and/or loads a potentially large amount of information to or from Riposte in bulk
Campus	One of two Horizon data centres in Bootle and Wigan. Each can handle the entire Horizon workload
Client	The client Financial Institution on behalf of which POL provides a service to Customers at Outlets
Cluster	Group of Correspondence Servers, all handling the same set of Outlets and replicating data between each other for resilience purposes
Cluster Lookup Service (LUC)	An NT service that supports Agents in a multiple Cluster environment. It provides, <i>inter alia</i> , a mapping of Riposte Groups to Clusters
Confirmation Agent	NBS Confirmation Harvester Agent takes [C1] messages from Counters and feeds them to the <i>Data Reconciliation Service</i> (DRS)
Correspondence Server	Hardware platform that supports the Campus-based Riposte Message Service, and handles message replication to and from a group of Outlets
Counter	Counter PC installed in a Post Office Outlet
Counter Clerk	Person working in an Outlet and operating a Counter
Customer	A member of the public transacting, or seeking to transact, business with POL through any of the Services
Data Reconciliation Service	Service provided by Fujitsu Services to POL which matches transaction flows from Counter and NBE, and reports on these to POL
Decline	Verdict supplied by the FI or by the Counter Clerk, such that a request for a withdrawal (or deposit) of funds is denied

EPOSS Settlement	Settlement of a Customer Session at the Counter
Fallback	Where a system has attempted to go On-line but failed and has the ability to proceed with the transaction in an Off-line manner – typically with limits on the transaction value
Gateway PC	Counter that has an attached ISDN line or other communications channel that enables it to pass data to and from the Campus
Generic Agent Server	Hardware platform that support the Agent processes for most applications (i.e. all those for which there are no application-specific requirements)
Handshake	Also known as an Echo Test. A message exchange between an NBX Authorisation Agent and an FI used to determine whether the remote application is capable of responding. Primarily used only when the TCP/IP connection is otherwise idle
Heartbeat	A message used to co-ordinate the Agent instances in a Resilient Agent Pair. For the NBX Routing Agent, this is a Riposte message, for the NBX Authorisation Agent it is a row in a table in the NBX Persistent Store
Heartbeat Table	The table in NPS used to hold NPS-based Heartbeats, the latest one from each Agent instance
Heartbeat History Table	The table in NPS used to hold the history of Heartbeats, one for each material change
Horizon	Name that encompasses the totality of the systems provided by Fujitsu Services to support the automation requirements of Post Office Outlets
Interactive Agent	Agent that handles an on-line message from a Counter, and returns an immediate response
Key Management	Covers the tasks involved in the generation, distribution and revocation of keys used for encryption
Logical FI	That portion of an FI with which an NBX Authorisation Agent interacts – also referred to as an FI_EE. To Horizon they are named CAPO_A, CAPO_B, LINK_A, LINK_B, AL_A and AL_B
Maestro	A proprietary scheduling system, produced by IBM (previously Unison Software). Used to schedule the Horizon Campuses. Now renamed Tivoli Workload Scheduler (or TWS); batch job scheduling and monitoring subsystem used to provide automated scheduling facilities within the Horizon system
Message Authentication Code (MAC)	Used to protect the integrity of the encrypted PIN value between PIN Pad and the NBX Authorisation Agent Server
NBX Authorisation Agent Server	Hardware platform on which the NBX Authorisation Agents run. It includes a HSM that is used to handle PIN transaction actions
NBX Routing Agent Server	Hardware platform on which the NBX Routing and Guaranteed Reversals Agents run
Network Banking Engine	A central system supplied by a third party and being replaced by NBX
Outlet	Former name for a <i>Branch</i>
Partner	The other Agent instance in a Resilient Agent Pair
PIN Blob	Block of data generated by a PIN Pad that contains not only the encrypted PIN value but also other data to protect its integrity
PIN Block	Block of data containing an encrypted PIN value
PIN Pad	Hardware device that is attached to a Counter PC and used by the Customer to enter a PIN value to authenticate a financial transaction

Platform	An instance of a hardware unit (server, workstation, Router etc) that is installed by and configured by POA to meet the Horizon security, application and capacity requirements
Priority Message	A Riposte message with a parameter that causes an immediate ISDN call to or from the Outlet, if the line is not currently open
Process Interface	The processing component in an FI
RAC Model	Basic model for banking transactions requiring on-line authorisation by an FI, where initial On-line Request [R] from Counter elicits On-line Authorisation [A] from FI. Confirmation [C] of outcome of transaction is sent in near time from Counter to NBE
Receipt	A printed record of the Transaction at the Outlet
Reconciliation	Ensuring the financial integrity of transactions across service boundaries
Reference Data	This is used in three different ways: <ul style="list-style-type: none"> • The end to end service for the receipt, manipulation and delivery of configuration data and parameters for use by the rest of the system, within the Horizon Programme • Read Only Data defined in the Riposte Message Store providing sets of Collections and Objects used to configure the Outlet and define the business parameters to be used and followed in providing a Counter service • The entirety of read only objects within the system, whether in the Riposte Message Store or not, that configure the system in some way or provide soft parameters to system definition and use
Release	A documented and co-ordinated collection of software and/or data provided by Fujitsu Services to deliver POL Services, or to extend the infrastructure used to deliver these services
Request	Request message [R] sent On-line from Counter to FI initiating an NBS dialogue
Resilient Agent Pair	A pair of Agent instances in which one is acting as a hot standby for the other. They exchange Heartbeats to control failover
Reversal	A Transaction that nullifies a specific previous Transaction that has been completed (committed) in a previous Customer Session, subject to business rules (e.g. time limits, previous receipt)
Riposte	Proprietary product from Escher group that is used to (a) support the Counter PC user's desktop, and (b) to provide a speedy and reliable message replication process between the Counters in an Outlet and the Correspondence Servers at the Campuses. The term includes WebRiposte whenever the context admits
Service Boundary	Intersection between operational domains
Service Level Agreement	Agreement to provide a quantified and measurable standard, required for a specified POL Service
Standby Agent	Of a Resilient Agent Pair, the instance that is not actively processing the workload
Token	Generic name for magnetic swipe cards, smart cards or bar codes used to initiate a Counter transaction
Track 2	Second track of data held on the magnetic stripe on a magnetic swipe card
Transaction	A recorded and auditable instance of business activity, involving service provision or Stock movement across organisational or service boundaries
WebRiposte	A version of Riposte that supports additional web functionality. (This additional functionality is not relevant to the Agents described in this document)

0.5 Changes in this Version

Fujitsu Services **High Level Design Specification for Agents for NBX,
the NBE Replacement** **Ref: NB/HLD/017**
COMPANY-IN-CONFIDENCE **Version: 2.0**
 Date: 13/03/2006

Version	Changes
1.1	Changes and corrections made during implementation: <ul style="list-style-type: none">– Signature verification suppressed under certain conditions (CP3896)– Handling of stale messages changed: no longer logged, threshold reduced to 15 seconds, 'real-time' [C0]s also subject to a staleness check (PC0112451)– 50% threshold of operational connections before second-choice allocation of [R1]s to connections is attempted (PC0112176) Addition of performance information previously missing Other changes: <ul style="list-style-type: none">– For LINK, reject a new inward connection when one already exists, but subject to a delay to give the old connection time to die (PC011214, from S90)
2.0	Version for approval, incorporating a few minor changes as a result of the formal review

0.6 Changes Expected

Changes
None

0.7 Contents

0.7.1 Table of Contents

0.	DOCUMENT CONTROL.....	2
0.1	DOCUMENT HISTORY.....	2
0.2	REVIEW DETAILS.....	2
0.3	ASSOCIATED DOCUMENTS.....	3
0.4	ABBREVIATIONS/DEFINITIONS.....	5
0.4.1	Abbreviations.....	5
0.4.2	Definitions.....	8
0.5	CHANGES IN THIS VERSION.....	12
0.6	CHANGES EXPECTED.....	12
0.7	CONTENTS.....	13
0.7.1	Table of Contents.....	13
0.7.2	Table of Figures.....	18
0.7.3	Table of Tables.....	19
1.	INTRODUCTION.....	21
2.	SCOPE.....	21
2.1	EXCLUSIONS.....	21
3.	DESIGN PRINCIPLES.....	22
3.1	ASSUMPTIONS.....	22
3.1.1	General.....	22
4.	REQUIREMENTS.....	23
4.1	KNOWN LIMITATIONS.....	33
5.	SYSTEM COMPONENTS.....	34
5.1	APPLICATION COMPONENTS FOR RAC AGENTS.....	34
5.1.1	Introduction.....	34
5.1.1.1	The RAC Model	34
5.1.1.2	NBX Persistent Store	35
5.1.1.3	Topology of the NBX Agents	35
5.1.1.4	Configuration	38
5.1.2	NBX Routing Agent (NX_NQ_RTNG).....	41
5.1.2.1	Overview	41
5.1.2.2	Structure, Launch and Concurrency	42
5.1.2.3	Functional Description	45
5.1.2.4	Exception Handling	48
5.1.2.5	Performance and Scalability	48

5.1.2.6	Resilience	49
5.1.2.7	Configurability	53
5.1.2.8	Audit	55
5.1.2.9	Operational Summary	55
5.1.3	<i>NBX Authorisation Agents (NX_NQ_CAPO, ..._LINK, ..._AL)</i>	57
5.1.3.1	Business Functionality	57
5.1.3.2	Interfacing with the FI_EEs	66
5.1.3.3	Operator commands	77
5.1.3.4	Security	80
5.1.3.5	Structure, Launch and Concurrency	82
5.1.3.6	Exception Handling	84
5.1.3.7	Performance and Scalability	85
5.1.3.8	Resilience	88
5.1.3.9	Configurability	91
5.1.3.10	Audit	94
5.1.3.11	Operational Summary	94
5.1.4	<i>NBX Guaranteed Reversals Agent (NX_HV_GREV)</i>	96
5.1.4.1	Overview	96
5.1.4.2	Structure, Launch and Concurrency	96
5.1.4.3	Detailed Processing	96
5.1.4.4	Exception Handling	97
5.1.4.5	Performance and Scalability	97
5.1.4.6	Resilience	98
5.1.4.7	Configurability	98
5.1.4.8	Audit	99
5.1.4.9	Operational Summary	99
5.2	APPLICATION COMPONENTS FOR EXISTING AGENTS.....	100
5.2.1	<i>NBS Confirmation Harvester Agent (NB_HV_CONF)</i>	100
5.3	UNCHANGED AGENTS.....	101
5.3.1	<i>TPS Harvester Agent (T_HV_ALL)</i>	101
5.3.2	<i>OMDB Heartbeat Harvester (M_HV_OMDB_HB)</i>	101
5.4	INTERFACES PROVIDED.....	102
5.4.1	<i>Interface between NBX Routing and Authorisation Agents</i>	102
5.4.1.1	RSTS	103
5.4.1.2	ASTS	103
5.4.1.3	[R1]/[C0]	104
5.4.1.4	[A3]/[A4]	104
5.4.2	<i>Probe Interface to the NBX Authorisation Agents for LINK</i>	105
5.5	INTERFACES TO EXTERNAL COMPONENTS.....	106
5.5.1	<i>Interfaces to Riposte</i>	106
5.5.1.1	Riposte Attribute Grammar	106
5.5.1.2	NBX Routing Data	106
5.5.1.3	Heartbeats	107
5.5.2	<i>Interfaces to FI_EEs</i>	107
5.5.3	<i>Interfaces to Cryptography and Key Management Services</i>	107
5.5.3.1	NBX Crypto API DLL	107
5.5.3.2	Crypto API DLL	108
5.5.4	<i>Interfaces to Oracle</i>	109
5.5.5	<i>Interfaces to NPS</i>	109
5.5.6	<i>Interfaces to DRS</i>	110
5.5.7	<i>Interfaces to OMDB Host</i>	110
5.5.7.1	Interfaces for OMDB Heartbeat Harvester	110
5.5.8	<i>Interfaces to TPS Host</i>	110
5.6	DISTRIBUTED APPLICATION SERVICES.....	110
5.7	INFORMATION MANAGEMENT.....	110

5.8	NETWORKING SERVICES.....	111
5.9	PLATFORMS.....	111
6.	SYSTEMS MANAGEMENT.....	112
6.1	SYSTEMS MANAGEMENT OF AGENTS.....	112
6.2	AGENTS' ROLE IN SYSTEMS MANAGEMENT.....	112
6.2.1	<i>Heartbeats as a Source for Monitoring.....</i>	<i>112</i>
6.2.1.1	Heartbeats in Riposte.....	112
6.2.1.2	Heartbeats in NPS.....	112
6.2.2	<i>Statistics as a Source for Monitoring.....</i>	<i>112</i>
6.2.2.1	Statistics in Riposte.....	112
6.2.2.2	Statistics in NPS.....	113
6.2.3	<i>NT Events as a Source for Monitoring.....</i>	<i>115</i>
6.2.4	<i>NBX Transaction and Management Journals.....</i>	<i>117</i>
7.	APPLICATION DEVELOPMENT.....	119
7.1.1	<i>Testing options for volume and performance testing.....</i>	<i>119</i>
8.	SYSTEM QUALITIES.....	120
8.1	RESILIENCE.....	120
8.1.1	<i>Resilience to a failing Correspondence Server.....</i>	<i>120</i>
8.1.2	<i>NBX Routing Agent.....</i>	<i>120</i>
8.1.3	<i>NBX Authorisation Agents.....</i>	<i>120</i>
8.1.4	<i>NBX Guaranteed Reversals Agent.....</i>	<i>120</i>
8.2	PERFORMANCE AND SCALABILITY.....	120
8.2.1	<i>NBX Routing Agent.....</i>	<i>120</i>
8.2.2	<i>NBX Authorisation Agents.....</i>	<i>120</i>
8.2.3	<i>NBX Guaranteed Reversals Agent.....</i>	<i>120</i>
8.3	SECURITY.....	121
8.3.1	<i>Service Users.....</i>	<i>121</i>
8.4	POTENTIAL FOR CHANGE.....	121
9.	SOLUTION IMPLEMENTATION STRATEGY.....	122
10.	MIGRATION.....	123
10.1	MIGRATION FROM NBE TO NBX.....	123
10.2	MIGRATION OF INDIVIDUAL AGENTS.....	123

0.7.2 Table of Figures

Figure 1 – RAC Model and Data Flows for NBX.....	34
Figure 2 – Topology of NBX Agents.....	36

0.7.3 Table of Tables

Table 1 – Synopsis of requirements from the Conceptual Design.....	33
--	----

Table 2 – Routing Agent Performance Targets (transactions/sec).....	48
Table 3 – Configuration of NBX Routing Agent.....	55
Table 4 – Response_Code Values for Agent-Detected Failures.....	60
Table 5: Journal_Types in the Transaction Journal.....	64
Table 6 – Network Management (0800) messages: Agent capability.....	70
Table 7 – Operator commands to the NBX Authorisation Agents.....	79
Table 8 – Authorisation Agent Performance Targets (transactions/sec).....	85
Table 9 – Cryptographic timings for processing an authorisation transaction.....	86
Table 10 – Transaction Journal control parameters.....	92
Table 11 – STAN-generation control parameters.....	92
Table 12 – Configuration of NBX Authorisation Agents.....	94
Table 13 – Registry for NBX Guaranteed Reversals Agent.....	99
Table 14 – Additional attributes harvested by the NBS Confirmation Harvester Agent.....	100
Table 15: Message types in inter-agent protocol.....	102
Table 16: Routing Agent Status (RSTS) message.....	103
Table 17: Authorisation Agent Status (ASTS) message.....	104
Table 18: Request messages in inter-agent protocol.....	104
Table 19: Response messages in inter-agent protocol.....	105
Table 20: AWK management functions by protection domain.....	108
Table 21: NPS tables accessed by NBX Agents.....	110
Table 22: Routing Agent statistics.....	113
Table 23: Statistics required for OMDB.....	115
Table 24: Other statistics reported to OMDB.....	115
Table 25 – Monitor Ids of the Resources for the NBX Routing Agent.....	116
Table 26 – Monitor Ids of the Resources for the NBX Authorisation Agent.....	116
Table 27 – Monitor Severity Levels.....	116
Table 28 – NT Events for Monitoring the NBX Routing Agents.....	117
Table 29 – NT Events for Monitoring the NBX Authorisation Agents.....	117
Table 30 – Service Users for NBX-specific Services.....	121

1. Introduction

This document is the **High Level Design (HLD)** for the new and changed Agents produced for the Horizon **NBE Replacement (NBX)** developments. It is an internal Fujitsu Services document.

These Agents are required for Release S75. The document has been updated for release S82R. It also includes updates for S90 – these are always specifically identified as such.

2. Scope

This document describes the high-level design for the new and changed Agents for NBX. The level of detail is intended to act as a baseline to Fujitsu Services (Post Office Account) developers and testers.

The system design for NBX is given in the Design Proposal for EMV, TDES and NBE Replacement [DP_NBX].

The system design defines a **Request/Authorise/Confirm (RAC)** model for Network Banking transactions. This is illustrated in Figure 1. New Agents are required to support this model:

- **NBX Routing Agent**, for routing [R1] and [C0] messages to the appropriate NBX Authorisation Agent
- **NBX Authorisation Agents**, enquiry agents for handling [R], [A], [C0] and [E] messages. There is a separate flavour of NBX Authorisation Agent for each of the FIs:
 - **CAPO Authorisation Agent**
 - **LINK Authorisation Agent**
 - **A&L Authorisation Agent**
- **NBX Guaranteed Reversals Agent**, for assured harvesting of [C0]s

Other Agents affected by NBX are:

- **NBS Confirmation Harvester Agent**, for harvesting of [C1]s to the DRS as [C12]s

2.1 Exclusions

The following are excluded from this design:

- None known

3. Design Principles

So far as is practicable, all new Agents should be designed in line with the existing generic models, structures and standards for Harvester, Loader and Enquiry Agents.

The new NBX Routing Agent should be modelled very closely on the existing NBS Authorisation Agent. Some aspects are better derived from the existing ETS Authorisation Agent, in particular the use of a generic format for the Heartbeat messages.

The new NBX Authorisation Agents should follow the same general structure as the existing Authorisation Agents. There are significant differences, in particular non-use of Riposte and the use of an NBX Persistent Store.

One of the requirements in the CD is "Each Interface {split by FI} must be designed to operate in a stand-alone manner, and must be in no way coupled to the interface for another FI." The different variants of the NBX Authorisation Agents, therefore, will result in separate executables and services.

Another of the requirements is that much of the configuration of the NBX Authorisation Agents must be "soft". This particularly applies to timers, such as the time of the end of the business day, and to protocol mappings, such as the mapping from FI response codes to Horizon response codes. The design principle should be that as much as is reasonable and practicable should be configurable.

Whenever an existing Agent has to be amended, it should be amended in such a way that existing functionality is untouched as far as is practicable. This is to avoid introducing the new bugs that would be introduced were the Agent to be unnecessarily re-engineered.

3.1 Assumptions

3.1.1 General

- NBX identifies a Network Banking transaction by a combination of the Group and Node Ids of the Counter that generated it, the last 6 digits of the message number component of the Horizon_Txn_Num, and the Receipt_Date (not time). NBX assumes this provides uniqueness (within 10 years, as only one year-digit is employed).
- The Agents need not do any character set transformations when sending messages to and receiving messages from the FI_EEs. Everything is in ASCII.
- The use of the Tertiary Bitmap field is not supported if the AIS defines that Bitmaps are transferred in binary. (*None of the current AISs requires this combination of support.*)
- When the Network Banking Counter Application writes the [R1] as a priority message, it does so in such a way as to keep the network connection open long enough for the [A3] to be returned on the same connection.
- LINK (as Settlement Master) change their Business Day at 8.00 pm, the same time as NBX changes its Business Day.

4. Requirements

The requirements affecting Agents are captured in the Conceptual Design [CD_NBX]. These are subject to the caveats in [CAVEATS].

The following table is a synopsis of the requirements, and the reader will need to refer to the Conceptual Design for a full statement of the requirements. However, the caveats are quoted in full. The table includes a very brief statement of how the requirement is met.

Ref	Synopsis of requirement	How met
NBR0298	NBX will be parameter driven, as defined by the NBX Business Parameters [BUSPARAMS].	NBX Configuration Parameters table in NPS.
NBR0298a	Changes to parameters shall at most require an outage of the service of less than 60 seconds.	Version control on the NBX Configuration Parameters table. Restart NBX Authorisation Agent(s) using a different version number.
NBX0060	The NBX will hold prior (current - 1) configuration of system parameters as a minimum and it will be possible to return to that configuration within the time stated in NBR0298a.	Restart NBX Authorisation Agent(s) using previous version number.
NBX0115i	NBX shall be resilient.	Permeates the design.
NBX0061	NBX should provide suitable de-coupling of functions.	Separate NBX Authorisation Agent service and executable and separate Configuration Parameters for each FI.
NBR0245	High availability as a design target.	Permeates the design.
NBR0622b	The online system must be able to handle error situations such as corrupt bitmaps, unreadable data etc.	May cause disconnection and/or 0620 Reject message
NBR0228	Data integrity across all interfaces.	Intrinsic to the design.
NBR0453	All data must be recorded at source. Likewise, Data should be recorded prior to leaving the Horizon domain.	Audited in the Transaction Journal.
NBR0498	Migration, including from NBE to NBX.	Controlled by Routing Data.
NBR0502	Migration reversion, including from NBX to NBE	Controlled by Routing Data.
NBR0277	Basic switching of messages.	Intrinsic to design.
NBX0003	Interfaces to the FIs will comply with the appropriate baselined AISs, TISs.	Intrinsic to design.
NBX0004	Further aspects of compliance with AISs	Intrinsic to design.
NBX0005	Each Interface {split by FI} must be designed to operate in a stand-alone manner, and must be in no way coupled to the interface for another FI.	Separate NBX Authorisation Agent service and executable for each FI.
NBX0063	The A&L, Card Account and LINK interfaces must be separate	Separate NBX Authorisation Agent service and executable and separate Configuration Parameters for each FI.
NBX0006	Each Application interface must be aware of the Networking Protocol.	The NBX Authorisation Agents use sockets and react accordingly.

**Fujitsu Services High Level Design Specification for Agents for NBX,
the NBE Replacement
COMPANY-IN-CONFIDENCE**

**Ref: NB/HLD/017
Version: 2.0
Date: 13/03/2006**

NBX0007	For the period of migration from the NBE to NBX, the Counter can reroute transactions.	Both the NBS and NBX Authorisation Agents use (the same) Routing Data to determine which Agents process which transactions.
NBX0008	All messages on an interface to a given FI work on the same Settlement Day. All physical connections to the FI (i.e. each PI) simultaneously move into the next Settlement Day.	The switch is controlled by the Agent's local clock at a configurable EOD cutover time.
NBR0218	Translate incoming messages from counter into the appropriate format for onward transmission to the FI.	The NBX Authorisation Agents map from [R1] and [C0] formats into [R3] and [E1] formats.
NBR0316	In the event of the NBX detecting that an FI interface has failed, it will decline message requests from the counter automatically.	When all PIs to an FI are 'Down', an appropriate failure-[A3] is returned.
NBX0064	Must be capable of concurrently supporting more than one physical connection to the FI. Each connection may support multiple parallel sessions.	Support for socket concurrency within each PI.
NBX0065	The Application Interfaces must be resilient and flag interruptions to service and data errors (i.e. format or structure errors) to the Administrators for immediate investigation.	MONID event messages for interruptions to service. Statistics on data errors harvested by OMDB.
NBX0066	Fujitsu must actively monitor and proactively recognise system degradation before it becomes service affecting.	Statistics generated by the NBX Authorisation Agents are interpreted by OMDB.
NBX0067	The NBX will act as the Settlement Master when in communication with A&L and Card Account and will act as the Settlement Slave when in communication with LINK	NBX Authorisation Agents are configured accordingly.
NBX0068a	Counter should supply all the transaction data.	Agents do not enrich the transaction data; they merely transform it. They do not even doing the FAD look-up to address.
NBX0068b	Minimise disruption to service. Caveat states "A constraint of the FI's is that on key exchange the FI's re-cycle the PI's, and the time for this to occur is not under the control of Fujitsu Services."	Intrinsic to design.
NBR0177	NBX will provide a consistent interface to the Counter.	Includes configurable translation of FI response and reversal reason codes.
NBR0219	Response message from FIs will need to be converted by the NBX to the Horizon format and vice versa.	The NBX Authorisation Agents map from [A1] format into [A3] format. The "vice versa" in the requirement appears meaningless.
NBR0288	Interface to LINK for balance enquiries, withdrawals, deposits and reversals.	LINK version of the NBX Authorisation Agent.
NBR0289	NBX will communicate to LINK using the LIS5 Message Standard.	The LINK mapping document conforms to LIS5.
NBR0299	NBX will communicate with LINK using TCP/IP.	The LINK Authorisation Agent uses sockets over TCP/IP.

NBX0069	The interface to LINK will be based on the LIS5 2004-1 specification.	The LINK mapping document conforms to this version.
NBR0302	Behaviour of Echo Tests for LINK.	The behaviour is as required except that the retry frequency should be faster than the regular Heartbeat frequency rather than slower. This is to establish as soon as possible whether or not an application session is faulty or has recovered. The LINK interface will be configured <u>not</u> to disconnect the TCP/IP connection when no Heartbeat responses are received.
NBR0305	Interface to A&L and CAPO.	A&L and CAPO versions of the NBX Authorisation Agent conform to relevant AIS and TIS documents.
NBR0305a	These interfaces should be developed in a generic manner. Caveat states "Additional direct interfaces to new FI's will be managed under the Change Control Procedure."	This is the design strategy.
NBR0423	NBX will interpret Response Codes received from each external on-line Interface and translate into the Horizon equivalent Response Code.	Performed using configurable mapping tables.
NBR0441	Audit trail. Caveat states "The audit trail will be maintained as described in the CCD entitled "Service Description for the Security Management Service" (CS/SER/016)."	Appropriate records audited in the Transaction Journal in the NPS (but with sensitive data omitted).
NBX0016	Transaction details in TES.	Transaction Journal captures all relevant transaction details for TES.
NBX0071a	Timestamps and elapsed times in TES.	Transaction Journal captures all relevant timestamps and elapsed times for TES.
NBX0115a	Elapsed times in TES.	Transaction Journal captures all relevant elapsed times for TES.
NBX0115f	Late Reversals in TES.	Late Reversals are flagged in the Transaction Journal.
NBX0114a	High availability of service.	Permeates the design.
NBX0114d	Limits on number and duration of outages per PI per month. Caveat states "The option specified in CT191 will allow for achievement of the requested SLA/OLA for the LiNK interface."	Avoidance of all unnecessary Logons, including avoidance of failover to standby Agent due to connection problems.
NBX0076	The NBX will allocate the Settlement Day to each message as defined in the Business Parameters document and the relevant AISs.	NBX's Settlement Date is changed in accordance with the configurable EOD Cutover time. (Also see NBX0008.)
NBX0114b	Report of agent PI Logons.	The NBX Authorisation Agents record Logons in the Management Journal in NPS for OMDB.
NBX0114c	Report of agent availability.	The NBX Authorisation Agents record relevant events in the Management Journal in NPS for OMDB.

Fujitsu Services High Level Design Specification for Agents for NBX, Ref: NB/HLD/017
the NBE Replacement Version: 2.0
COMPANY-IN-CONFIDENCE Date: 13/03/2006

NBR0179	Stale requests.	The NBX Authorisation Agents audit stale [R1]s in the Transaction Journal for TES. <i>(This audit was subsequently abandoned to maintain the stability of the Agent under stress – see 5.1.3.1.1)</i>
NBX0116f	Reports on response times.	The NBX Authorisation Agents record the raw elapsed times in the Transaction Journal for TES.
NBX0021	NBX must check for duplicate requests, must be aware if FI is down, update the transactions status, set a timer on the [A1] response.	Included in the design.
NBX0096	Handling of normal [A1]s, [A1] timeout and late [A1]s.	Included in the design.
NBX0097	Handling of [C0] reversals. If not a duplicate and not declined or already reversed, then forward to the same PI. A reversal message is a must deliver message.	Included in the design. For CAPO, have implemented the stronger requirement that the reversal is sent to the same remote IP address and port (i.e. to a specific port within the PI) as per the TIS. NBX Guaranteed Reversals Agent ensures that every [C0] reversal message is processed. The NBX Authorisation Agents ensure that it is forwarded and retried as required. The Transaction Status is retained for 5 days. A [C0] reversal received from a counter more than 5 days after the original transaction is audited to the Transaction Journal but not processed.
NBX0098	NBX must be able to accept and recognise and generate status and error codes in line with the TCP specification.	The acceptance criteria should not be in terms of TCP/IP error codes (the TISs do not contain such a list).
NBR0004	Support for Balance Enquiries.	Supported.
NBR0325	Support for multiple balances as per AISs and Mapping Documents.	Supported. For A&L the Authorisation Agent will process the balances correctly no matter in what order they are returned. For CAPO and LINK the balances are returned in the correct order.
NBX0099	Absence of balances.	Handled by the Mapping Documents.
NBX0100	Mapping of Response Codes.	Mapped as per the relevant Mapping Document.
NBR0002	Support of Cash and Cheque Deposits for which no Customer Verification is required.	Supported.
NBR0002a	NBX shall be generic in the handling of Customer Verification data.	Point of Service Entry Mode, Point of Service Condition Code, and Cardholder authentication method in Point of Service Data are mapped. The PIN and ICC Data fields are derived dynamically from the presence of that data in the [R1] message.
NBR0003	Support for Cash Withdrawals.	Supported.
NBR0623	Support for [C0] reversals.	Supported (see NBX0097).
NBX0101	Support for listed transaction messages.	Supported in accordance with AIS and TIS documents.

Fujitsu Services High Level Design Specification for Agents for NBX, Ref: NB/HLD/017
the NBE Replacement Version: 2.0
COMPANY-IN-CONFIDENCE Date: 13/03/2006

NBR0158	Transactions will be uniquely numbered as generated by the counter application, however, these numbers will not necessarily be incremented by one each time.	The Trans_Num identifier used is the last 6 digits of the message number component of the Horizon_Txn_Num attribute. This number is unique within well over a week.
NBR0178	Stale messages will be logged but not passed on to the FI.	Period is configurable through the NBX Configuration Parameters in NPS. <i>(This audit was subsequently abandoned to maintain the stability of the Agent under stress – see 5.1.3.1.1)</i>
NBX0022	The check for stale messages is not affected by any changes in System Time Base.	Check uses UTC, measuring from when the [R1] is read by the NBX Routing Agent.
NBX0023	[C0] reversals are not subject to the staleness check.	Yes, for [C0]s read by the 'guaranteed' route. However, [C0]s read by the 'real-time' route will be discarded if they are stale.
NBR0184	NBX will assign a Settlement date to all authorisations as defined by the relevant AIS.	Yes. (Also see NBX0008.)
NBR0227	Data errors identified are reported.	Suitable audit records logged to Transaction journal for TES.
NBX0024	Horizon-generated Response Codes.	Same values used as for when interfacing to NBE.
NBX0102	A problem with a message on an interface should not cause all other messages carried out subsequently to be queued.	Intrinsic to the design.
NBR0020	The system shall support the RAC (Request, Authorisation, Confirmation) model of End-to-End transaction flows.	Intrinsic to the design.
NBR0157	The receipt date and time will be carried through any dialogue with the Bank.	Used as the Date, Local Transaction and Time, Local Transaction fields (bitmap refs. 013 & 012) in the messages sent to the FI.
NBR0279	Elapsed time, recorded in milliseconds, for the time spent by the Authorisation Agent waiting for a response from the FI (which includes the time spent with the LINK Client bank).	Recorded as Agent_SLA_Info.
NBR0279a	Various specified transaction timestamps will be audited to the journal, using local system clock.	They are captured in Riposte_Tsmp and/or Agent_Event_Tsmp in various Transaction Journal records (but see NBR0279b).
NBR0279b	Exceptions to NBR0279a.	These are genuine exceptions.
NBR0536	NBX Agents will log various specified attributes of a transaction.	They are captured in various Transaction Journal records. The 'Transaction Sequence Number' is a combination of the Terminal Id (group and node) and the Retrieval Reference Number.
NBR0537	The source and destination of the transaction shall be recorded.	The source is captured in Terminal Id in various Transaction Journal records. The destination is recorded as the FI Type (but if 'destination' means the bank, this is recorded indirectly in the PAN).

NBR0023	Transactions at the counter shall be generic.	The 'NBX routing ID' is the Routing_Gateway indicator. The NBX Routing Agent is driven by reference data (Routing Data) to route the transaction to the correct NBX Authorisation Agent.
NBX0103	Table of supported transactions.	Reflected in NBX Configuration Parameters: TxnType and MsgId. Note that [C0] for a PIN Change does not result in an [E].
NBR0300	The NBX will decline [R] transactions and return the [A] to the outlet in the case of time-outs for an Authorisation being exceeded. The NBX will then reply to the FI should it respond with an approved authorisation by issuing a Reversal Request to reverse the Authorisation.	Supported. (See also NBX0096.)
NBR0301	The NBX will allow time-outs to be configured for the window in which a message response from a FI will be allowed.	The timeout used for timing out an [A1] is the value from the [R1], controlled by the Reference Data item MAAWP (Maximum Authorisation Agent Wait Period), in accordance with the Contract.
NBX0053	Clock synchronisation.	Agents use the local system clock. These are synchronised across the data centre servers using the Time service.
NBX0054	Clock synchronisation. Caveat states "Fujitsu Services will use BST in the day light saving period and GMT otherwise, for all FI facing components for the provision of transmission date and time fields."	Agents use the local system clock. These are synchronised across the data centre servers using the Time service.
NBR0632	[A1]s and timeouts.	Supported. (See also NBX0096.)
NBX0025	The interface code must be able to be run in parallel to cater for load increases, and further instances must be instantiated to cater for additional load and new functions. Caveat states "The NBX solution has been designed to support scalability of Agent platforms, as is demonstrated in S75 with two instances of each FI's Agent. Any further Agent instances will be handled by the Change Control Procedure."	The solution is designed to support scalability as stated in the caveat.
NBX0026	Layering of the solution.	Layered where appropriate.
NBX0055	Recovery after failure or outage.	Orderly recovery and data integrity after failure permeate the design.
NBX0120	Resilience to failure. Caveat states "Upon failure of one element of the NBX it cannot be guaranteed that the service will not fail-over without end user perceived degradation."	Use of standby components permeates the design.
NBR0153	Use of Euros.	Currency support and mappings for NBX Authorisation Agents are supplied by NBX Configuration Data.

Fujitsu Services High Level Design Specification for Agents for NBX, Ref: NB/HLD/017
the NBE Replacement Version: 2.0
COMPANY-IN-CONFIDENCE Date: 13/03/2006

NBR0318	NBX solution must be Euro compliant.	As for NBR0153.
NBR0569	Between Horizon and the FIs messages will be sent as per the agreed interface specification (AIS/TIS) and any security protection will be as stated therein.	Yes.
NBR0570	The current NBE/Horizon MAC will no longer be required, but the PIN MAC for on line PIN will remain.	NBX Authorisation Agents invoke the necessary PIN block translation.
NBR0575	PINs shall be protected in accordance with the requirements of each relevant AIS.	NBX Authorisation Agents invoke the necessary PIN block translation.
NBR0576	PIN Block formats shall comply with the requirements of each relevant AIS.	NBX Authorisation Agents invoke the necessary PIN block translation.
NBR0577	PINs shall never appear in plain text other than within a tamper-resistant hardware security module or a tamper-evident PIN Pad as defined by the standard.	NBX Authorisation Agents invoke the necessary PIN block translation.
NBR0588	Fujitsu Services Ltd shall specify what enforced disconnection facilities they consider necessary.	The operator interface to the NBX Authorisation Agents provide facilities to disconnect a PI.
NBX0114k	Zone Master Key changes should be carried out in line with the requirements of each Interface, with a total service outage of no more than 90 seconds.	Necessary facilities are supported by the NBX Authorisation Agents.
NBR0534a	Fujitsu Services will monitor the system and produce alerts for error states.	The NBX Authorisation Agents provide the feeds required by OMDB, etc: MONID event messages, Heartbeats, statistics, Management Journal in NPS.
NBR0535a	Provision of 'NBX "Console" Interface':	The NBX Authorisation Agents react to commands via an NBX Operator Commands table in the NPS.
NBR0535b	... for Logon and Logoff messages.	Supported.
NBR0535c	... for forcing manual (re) transmission of Network Management message on any interface.	Supported.
NBR0535d	... for forcing manual removal and logging of unsuccessfully transmitted store & forward transactions.	Supported by specifying a time range embracing the affected reversals.
NBR0535e	... for manually sending transactions that may be "stuck" on a store and forward queue.	Supported by specifying a time range embracing the affected reversals.
NBR0535g	The solution should provide a user readable real-time view of events taking place on the FI facing component of the NBX, and maintain a log to assist investigation. This log shall contain trace details and be exportable.	The NBX Authorisation Agents record events in the Transaction Journal for TES.

NBX0114h	Where an FI failure is noted that affects the service e.g. an unplanned outage of a FI PI at the FI end, the load should be shared as defined in the TIS and the duration of such support of the FI failure must be recorded. No associated SLA.	The load is shared as defined, though because an NBX Authorisation Agent is multithreaded there is a small but variable delay between allocating an [R3] to a particular operational PI and the subsequent attempt to transfer it to that PI. The NBX Authorisation Agents record relevant events in the Transaction Journal for TES.
NBX0114j	Where an element, or elements, of the Fujitsu Services infrastructure fails that impacts the NBX service, and either the resilient nature of the architecture takes over the processing, or an external service effect is noticeable, this must be recorded.	OMDB's monitoring and recording of the Heartbeats meet this requirement.
NBR0612	The service must be supported by active monitoring of the system and proactive management. This must include preventive work as well as monitoring and investigating behaviours arising in operational service or technical areas. Caveat states "With regard to active management where no incident of problem has been raised the Service Management Forum will provide forward analysis and decision making in relation to the NBX as is currently the case for NBS. There will be no change to the provisions of the Agreement or to CCD's."	The NBX Authorisation Agents provide the MONID event messages, Heartbeats, statistics, etc, as raw data for such monitoring.
NBR0016	All transactions must be fully auditable.	The NBX Authorisation Agents log all relevant audit points to the Transaction Journal for audit purposes.
NBR0016a	All data associated with a transaction within the TES will be available to be returned with the transaction record detail (except for the PIN contained within the PIN Block and Track 2 / Track 2 Sensitive Data).	The NBX Authorisation Agents log all allowed data to the Transaction Journal for TES.
NBX0051	All systems will log user activity. Any changes made to the system must have the user name of the user making the change associated with it, coupled to the date / time of the change.	Operator commands through NBX bear such information and this is logged to the Management Journal.
NBR0629	The NBX Service continues to operate with full processing capacity should the service at one site fail, and that all transaction data is available at the second site.	The transaction data is made available through the Transaction Status and C0 Reversals tables in the NPS. Note that the failover to the standby Agent at the other site is not automatic if the failure concerning the communications interface with the FI.
NBR0630	The NBX Service continues to operate with full processing capacity should the communications to one site fail.	Note that the failover to the standby Agent at the other site is not automatic if the failure concerning the communications interface with the FI.

NBX0035	The [R]/[A] interface for a given FI must be closely coupled to the reversal process for that FI such that if the reversal interface fails the corresponding [R]/[A] interface should be suspended immediately.	The NBX Routing Agent routes a [C0] reversal to the same 'logical' NBX Authorisation Agent as the original [R1] using a shared transfer mechanism. The same NBX Authorisation Agent instance handles both the [R]/[A] and reversal interfaces for a given FI.
NBX0036	An outage of an individual PI to an FI must not isolate any part of the Branch network from undertaking transactions.	The TISs require that each NBX Authorisation Agent instance support at least two PIs. Failure of one PI causes [R3]s to be assigned to one of the other PIs.
NBX0037	The NBX solution should contain no single points of failure. Recovery from failure of a component involved in supporting the online service should be automatic; with the exception of the NPS which under disaster scenarios may require manual recovery actions.	Standby Agent instances for both the NBX Authorisation Agents and the NBX Routing Agents. NBX Routing Agents access two Correspondence Servers. NBX Authorisation Agents access the NPS via two Oracle instances. The NBX Guaranteed Reversals Agents connect to whichever Oracle instance is available. Note that the failover to the standby Agent at the other site is not automatic if the failure concerning the communications interface with the FI, but this would involve more than a single point of failure.
NBX0038	Excluding failures that require Disaster Recovery, failure of any on-line components within the NBX that handles Branch traffic should be detected within 30 seconds. The recovery of any component solely under Fujitsu Services control should take less than 30 seconds from point of detection. Hence a maximum of 60 seconds to recover. For the avoidance of doubt, the intention is that the transaction at the counter will succeed on its subsequent retry following its initial failure.	This requirement permeates the design of NBX Agents and other NBX components. Thus, each NBX Authorisation and Routing Agent has a hot standby instance at the other data centre ready to take over. The NPS is accessed via two Oracle instances, again with hot standby threads in the NBX Authorisation Agents already connected to the reserve instance ready to take over. Timeouts for detecting failure are set aggressively: e.g. detecting a missing heartbeat from the active NBX Authorisation Agents is set to 25 seconds, from the active NBX Routing Agent is effectively at most 16 seconds. Having detected the failure, the Agent undertakes recovery action as quickly as possible. For example, following the loss of a critical resource that requires failover to the standby agent instance, the active agent 'resigns' and the standby agent takes over within 5 seconds when it sees the 'resignation' in the heartbeat.
NBX0038a	Under normal operation the workload through the components directly interfacing with each FI shall be split across more than one data centre.	The 'A' and 'B' versions of the NBX Authorisation Agents for each FI will be configured to have their primary servers at different sites. The 'A' and 'B' versions each deal with approximately 50% of the Branch Network.
NBX0039	The status of any component within the NBX, and the "health" of each component should be reported to a real-time monitoring tool.	The NBX Routing and Authorisation Agents assist this monitoring by generating MONID messages on their health and the health of certain resources they use.

NBX0109	Demonstration of performance capability.	The NBX Authorisation Agents provide specific 'test' facilities to enable volume testing.
NBR0459	Performance capability. Caveat states "The volumes for network banking as stated in CCD entitled "Horizon Capacity Management & Business Volumes" (IPA/PER/033) will be updated to include Post Office Ltd projected Volume model comparison v.0.3 - Supplier Sheet only. The Capacity Management sub group of the Service Management Forum will approve the updated "Horizon Capacity Management & Business Volumes" document (PA/PER/033)."	The performance of the Agents permeates their design.
NBR0461	Full performance infrastructure from day 1.	The Agents are geared to the contracted transaction volumes from the start.
NBX0040	Monitoring capacity.	The NBX Routing and Authorisation Agents both provide raw statistics to aid such monitoring.
NBX0119	Migration from NBE to NBX and cut back.	Both the NBS and NBX Authorisation Agents use (the same) Routing Data to determine which Agents process which transactions. (Also see NBR0023.)
NBX0110	The NBX will support the transactions described in each FI AIS and TIS.	Supported. (Also see NBX0103.)
NBX0111	The NBX will support the events listed in Appendix D of [CD_NBX].	The NBX Authorisation Agent will have a single business parameter covering both 'NBX System Business Day' (at 19:59:59) and the relevant EOD events at 20:00:00. These two events are not logically distinct.
NBX0113	The NBX will carry out the functionality described in appendix F (Business Parameters) of [CD_NBX] or provide an equivalent to it. Note that appendix F states "As this describes current functionality, not all is necessarily appropriate. It has been left in its entirety so as not to pre-suppose any solution."	Most of this 'soft' configuration data is configured in the NBX Configuration Data in the NPS. The Horizon Routing_Gateway to internal routing code (Logical FI) is configured by Type D reference data.

Table 1 – Synopsis of requirements from the Conceptual Design

4.1 Known limitations

- The several TCP/IP connections to a PI may either be all to the same remote virtual address (IP address and port) or all to different addresses. The NBX Authorisation Agent has a limitation that it does not cater for several connections to each of several VAs.
- The NBX Authorisation Agent has a limitation that restricts the socket concurrency to one per PI for inward connections.

Figure 1 – RAC Model and Data Flows for NBX

The NBX Authorisation Agents use a persistent storage mechanism called the **NBX Persistent Store (NPS)**. The NPS includes the following:

- Transaction Journal for logging and audit purposes, serving as a feed to the Transaction Enquiry System (TES).
- Transaction Status tables for all Transactions for the last five complete business days. The information includes sufficient to resend 'must-deliver' Reversal messages to the FI.
- Transient table of [C0] Reversal messages harvested by the NBX Guaranteed Reversals Agents.
- Reference Data for certain configuration parameters, mainly business parameters, including for FI-specific mapping of response codes. Note: Response code mappings are FI-specific (not specific to an individual card issuer).
- Commands table for the operator (via Tivoli) to input commands to an NBX Authorisation Agent.
- Heartbeat information exchanged between active and standby instances of the NBX Authorisation Agents.
- Heartbeat history, raw statistics and a Management Journal as a feed to OMDB for systems management purposes.

5.1.1.3 Topology of the NBX Agents

The replacement of the NBE requires a new topology in terms of the Network Banking Agents. The Agents are split in two, with **NBX Routing Agents** facing the Correspondence Servers and **NBX Authorisation Agents** facing the external FIs. In addition, new **NBX Guaranteed Reversal Agents** use checkpoints to ensure the harvesting of all Counter-generated Reversals.

Figure 2 shows the NBX Routing and NBX Authorisation Agents to service the FIs (LINK, CAPO and A&L at S75). *(Note: It only shows 'active' Agents, omitting the 'standby' Agents needed for resilience; it also omits the NBX Guaranteed Reversals Agents.)*

The NBX Routing Agents deterministically route messages, without transformation, from the four Correspondence Server Clusters to the appropriate NBX Authorisation Agent, one or more for each FI.

At S75, there will be two NBX Authorisation Agents for each FI. The two for CAPO are for performance reasons, and they will therefore be hosted on separate platforms. Those for LINK and A&L are for resilience reasons, and so can share platforms.

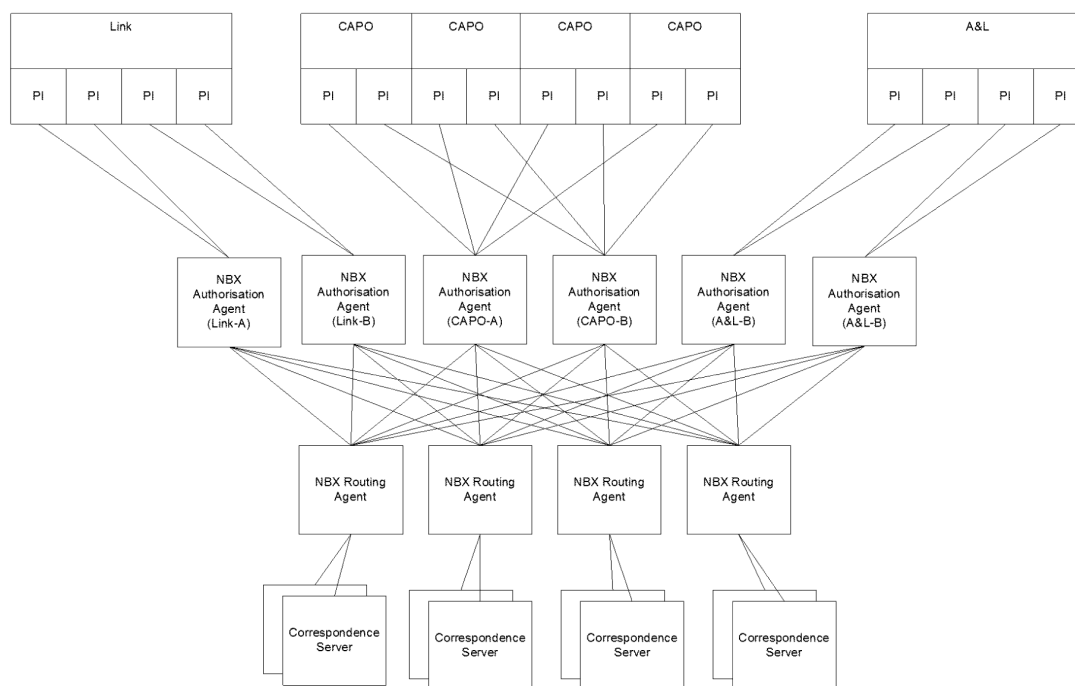


Figure 2 – Topology of NBX Agents

5.1.1.3.1 Topology of the NBX Routing and Authorisation Agents

There is one NBX Routing Agent and one NBX Guaranteed Reversals Agent per Correspondence Server Cluster.

There is one NBX Authorisation Agent for each Logical FI, i.e. for what this HLD terms an **FI Enquiry Engine (FI_EE)**. Thus at S75 there will be one Authorisation Agent for each of CAPO_A, CAPO_B, LINK_A, LINK_B, AL_A and AL_B.

For resilience, each NBX Routing Agent service is configured to connect to both Correspondence Servers at its local site and will at all times use whichever is available. Each NBX Guaranteed Reversals Agent service, on the other hand, is configured to connect at start-up to whichever local Correspondence Server is available, but it will not automatically switch to the other one should there be a problem. *(This is the same behaviour presently exhibited by the NBS Authorisation and Expedited Confirmation Agents respectively.)*

Similarly, each NBX Authorisation Agent service is configured to talk to both Oracle instances of the NBX Persistent Store (NPS) and will at all times use whichever is available. Each NBX Guaranteed Reversals Agent service, on the other hand, is configured to connect at start-up to whichever NPS Oracle instance is available, but it will not automatically switch to the other one should there be a problem.

Each 'logical' NBX Routing Agent operates as a Resilient Agent Pair, in which one service instance is acting as a hot standby for the active one. They exchange Heartbeat messages via Riposte to control failover. *(This is the same behaviour presently exhibited by the NBS and other Authorisation Agents.)*

Similarly, each 'logical' NBX Authorisation Agent operates as a Resilient Agent Pair, in which one service instance is acting as a hot standby for the active one. They exchange Heartbeat messages to control failover, but this time via the NPS.

The active NBX Routing Agents establish TCP/IP connections to each of the target NBX Authorisation Agents for sending [R1] and [C0] messages and for receiving [A3] (and exceptionally [A4]) responses. The Authorisation Agents listen on four ports, one per Cluster. This means that at S75 the six active and six standby Authorisation Agents collectively listen on 48 different ports.

The standby Routing Agents also establish these connections, so that there is no delay when a Routing Agent fails over to its standby. Conversely, the Routing Agents maintain connections with both the active and standby Authorisation Agents (which means that at S75 each Routing Agent maintains a total of 12 connections).

Before any banking messages are transferred on such a connection, there is an exchange of status messages, whereby each of the Routing and Authorisation Agents declares whether it is active or standby. Furthermore, each Agent will send another status message whenever its status changes (from active to standby or *vice versa*). By this means, a Routing Agent knows which Authorisation Agents are active – it will send banking messages only to an active Authorisation Agent.

The design allows for each Routing Agent to open more than one connection to each Authorisation Agent. However, there does not seem to be any performance need to exploit this option, so the default configuration is for just one.

An Authorisation Agent will send the [A3] response back on the same connection on which it received the [R1]. (There is no response to a [C0].) If the connection has been lost and replaced, it will send it back on the replacement connection, failing that it will send it back on any connection serving the same Cluster (except for a Ping connection), even if this is to the standby Routing Agent. Thus, when the active Routing Agent is failing over to its standby, the response is returned to whichever Routing Agent has an open connection at that moment. If there is no suitable connection, the response remains on its queue waiting to be sent.

The Authorisation Agent can send a 'hold' status message to a Routing Agent, to instruct the Routing Agent not to send it any banking messages for the present. Banking response messages in the opposite direction are not affected by the hold. The Authorisation Agent may issue the hold instruction when it would have problems with processing the message. Its main use is intended to be when the Authorisation Agent is about to fail over to its standby, so that Routing Agent can temporarily hold the messages and shortly route them to the newly active Authorisation Agent.

(Whilst the Routing Agents support the 'hold' concept, the Authorisation Agents will not exploit this facility at S75 (see 8.4 Potential for Change.)

The Routing Agents, both active and standby, will when necessary 'ping' an Authorisation Agent. To do this they send a 'ping' status message on a fresh connection to the same listening port, closing the connection again afterwards. Such Ping connections are not used for transaction messages. The prime purpose of pings is when an active Routing Agent suspects a problem communicating with an Authorisation Agent and negotiates with its standby as to whether the standby Agent can offer a better service. *(The NBX Routing Agent uses the same failover strategy and behaviour presently exhibited by the NBS and other Authorisation Agents. Designing the Agent from scratch might have used an alternative mechanism to pings.)*

The message protocol between the NBX Routing and Authorisation Agents is defined in 5.4.1. The protocol includes two fields, the Cluster Id and the name of the Logical FI, that enables the Authorisation Agent to detect most examples of misconfiguration.

5.1.1.3.2 Topology of the NBX Authorisation Agents and the FIs

Each Logical FI comprises two or more distinct **Processor Interfaces** (PIs). Each PI is essentially self-contained. With CAPO, each PI comprises two distinct 'threads'. These remote entities (PIs for LINK and A&L, threads for CAPO) can be considered to be the 'unit of processing', each with its own **Virtual Address** (VA, an IP address and port number).

Load balancing of [R3] messages operates across these 'units'. [E1] Reversal messages have to be sent to the same 'unit' as the [R3] being reversed.

The Technical Interface Specifications (TISs) define whether the Agent or the FI_EE's PI initiates the TCP/IP connections. NBX initiates the connections with CAPO and A&L; LINK initiates the connections to NBX. The TISs also define the number of connections per PI and the use of Virtual Addresses. The several TCP/IP connections to a PI may either be all to the same VA or all to different VAs. The NBX Authorisation Agent has a limitation that it does not cater for several connections to each of several VAs.

With inward connections, NBX offers a single virtual address per PI for the FI_EE to connect to, no matter at which campus the active Agent is running. The NBX service is virtualised using a Content Switching service (see the LINK TIS and the NBX Network HLD). The Content Switch probes the Agents at both campuses to determine which one is the active instance and which the standby, and routes inward connections from the FI_EE to the active instance. The active Agent listens on a special port configured for the health probe, the standby Agent does not. It is this difference in behaviour that allows the Content Switch to determine which is which.

Network Management messages, including the use of cryptographic working keys, operate at the PI level. Log On messages are used to establish a **PI session**, with a new working key (AWK) agreed at session start. The two 'threads' in a CAPO PI share one PI session. The cryptographic system for NBX does not share AWKs between platforms, so a standby Agent is necessarily unable to inherit a PI session from an active Agent. The approach adopted is that a PI session is deemed to have become unavailable whenever the last (or only) TCP/IP connection fails or is disconnected.

Only the active Authorisation Agent of a Resilient Agent Pair forms connections with the FI_EE. The standby Agent does not initiate outward connections, does not listen for inward connections from the FI_EE, and does not listen for health probes. The active Agent will not fail over to the standby Agent because of problems communicating with the FI_EE. The network is deemed to be sufficiently resilient and capable of reconfiguring itself that any problem will affect both Agent instances equally. (*Note: This behaviour is different from that of any previous Authorisation Agent.*)

See 5.1.3.2 for more detailed information on interfacing with the FI_EEs.

5.1.1.4 Configuration

The conceptual design states certain NBX parameters need to be soft and the following categories have been identified:

- Some parameters will be held as Registry parameters. Examples include the host and service names used for the connections to the FI, and parameters needed by the Agent during initialisation before it accesses the NPS (such as the number of threads of various types).
- Some parameters will be held as Type D Reference Data in the Riposte message store. Examples of this data are: NBX Routing Data.
- Some Reference Data parameters will be held as NBX Configuration Parameters in the NPS. Examples of this data are: the time of the end of the business day, the mapping from FI response codes to Horizon response codes.
- Some parameters are soft within the Agent source code, but will effectively be hard coded once the system has been tested and deployed. These are held in the NBX Configuration Parameters table in NPS. Examples of this are protocol mapping parameters for specific FI's. (*Note that only the NBX Authorisation Agents are configurable by this method.*)
- Some parameters are Branch specific Reference Data, supplied by the Counter software within the [R1] message. An example of this is the short form of Branch address passed on the [R3] message.

The NBX Authorisation Agent Reference Data and registry parameters will only be read on start-up and therefore will not be dynamic. The procedure to move the Agent to use a different set of parameters will be as follows: update the single system parameter, stop the standby Agent and let it restart with the new version of configuration parameters, then stop the active Agent and fail over to the standby agent – which would typically involve an outage of less than 30 seconds. When the formerly active agent restarts as the new standby agent, it will also use the new version of configuration parameters.

5.1.1.1.1 NBX Routing Data

For the NBE Replacement at S75, new Routing_Gateway values are being introduced for the three directly connected NBX interfaces: A&L, LINK and CAPO. Transactions to be processed by the new NBX Agents will use these new Routing_Gateway values. Those to be processed by the obsolescent NBS Agents and NBE will continue to use the existing Routing_Gateway values.

From S70 both the NBS Authorisation and Expedited Confirmation Agents will be capable of using the **NBX Routing Data** on which this decision is taken. The configuration data itself need not be deployed until S70R. These NBS Agents will process any Transaction that is not explicitly configured to be processed by NBX.

The criteria on which the new NBX Routing and Guaranteed Reversals Agents and the revised NBS Confirmation Agent will make its routing decision will be:

- Routing Gateway
- Cluster number
- Agent_Hash value

Because of the impact on Reversals, changing the criteria can only be done through the Change Control Process.

See 5.5.1.2 for more information on NBX Routing Data.

5.1.1.1.2 NBX Configuration Parameters table in NPS

The general principal is that the NBX Configuration Parameters table (the **TMS_TX_NBX_CONFIGURATION** table) in NPS is for configuration data that would have been Reference Data on traditional Agents.

One specific principle is that configuration via the NPS is at the level of a specific FI Type (i.e. for CAPO, LINK or A&L). It does not provide for any lower level of granularity, for example by logical FI (e.g. CAPO_A as distinct from CAPO_B) or by NT service name.

Each row in the table is for a specific configuration item, identified by its Parameter_Name, and Applies_To a specific FI Type. Rows with any other Applies_To value act as ‘comments’ and provide a description of a specific configuration item or set of configuration items; by convention, an Applies_To value of “Description” is used for such rows.

For documentary purposes only, an Applies_To value of “Generic” is used for configuration items that apply equally to all FI Types. However, such items will have been expanded into one row for each FI Type in the **TMS_TX_NBX_CONFIGURATION** table itself.

Those ‘business parameters’ that are subject to some level of change control between Post Office Ltd. and Fujitsu Services Ltd. are documented in [BUSPARAMS], which is a Contract-Controlled Document. The business parameters there are classified as to whether they can be changed by Operational Level Agreement or by Contractual Change Control.

[BUSPARAMS] excludes those parameters that Fujitsu Services Ltd. can change without prior consultation with Post Office Ltd., subject to there being no change in the characteristics of the service contrary to those contractually agreed and accepted. Such parameters are documented in this HLD – see 5.1.3.9.1.

5.1.2 NBX Routing Agent (NX_NQ_RTNG)

The Low Level Design document for this Agent is [NBXROUTING].

5.1.2.1 Overview

The **NBX Routing Agent** will listen for [R1] and [C0] messages through a Riposte real-time message port, add a time-stamp and route each message to the appropriate NBX Authorisation Agent. There will be one Routing Agent for each Correspondence Server Cluster. The use of a Riposte real-time message port ensures that the Agent will only process 'fresh' messages.

In the diagram of the RAC model, Figure 1, it will be seen that [R1] and [C0] input messages map on to [R1] and [C0] output messages, and that an [A3] input message maps on to an [A3] output message.

For clarity of exposition of the NBX Routing Agent, the messages between this Agent and the NBX Authorisation Agent have to be given distinct labels: [R2], [R2] and [A2] respectively. This choice of nomenclature accords with inherited conventions and hence is employed within the source code for the Agent.*

However, for the descriptions of the NBX Authorisation Agent and of the inter-Agent protocol, these messages are always referred to as [R1], [C0] and [A3] as per the RAC model.

The Routing Gateway Indicator is the primary criterion for routing the message to an NBX Authorisation Agent serving LINK, CAPO or A&L. There may be more than one Authorisation Agent for each such FI. Secondary criteria are the Cluster number and the Branch's Agent_Hash value.

The routing table is statically configured as Type 'D' Reference Data and ensures that a [C0] Reversal is routed to the same logical Authorisation Agent as the [R1] it is reversing.

New Routing Gateway Indicator values will be configured at the Branches for Transactions destined for the NBX. This enables them to be separated from NBE Transactions during reconciliation and in the DRS reports. The Routing Agent will ignore Transactions using unrecognised Routing Gateway Indicators, in particular those currently used by the NBE; this is as an aid to migration. Note: For the migration period, the existing NBS Authorisation and Expedited Confirmation Agents will be provided with the same routing table of the Routing Gateway Indicator values for NBX. They will ignore Transactions destined for NBX. Migration is more fully discussed elsewhere.

The messages are transferred from an NBX Routing Agent to an NBX Authorisation Agent by means of a permanently connected socket. Individual transfers are not acknowledged. Transfer failure of an [R1] will eventually result in the Counter timing out the Authorisation. Transfer failure of a [C0] is catered for by having an NBX Guaranteed Reversals Agent also harvesting the [C0] and committing it to the NBX Persistent Store. A Routing Agent will never route a message to a standby Authorisation Agent instance as well as to the active instance; there is never any split in the message flow.

The [A3] Authorisation response is transferred back to the same (logical) NBX Routing Agent that it received the [R1] from. If meanwhile the NBX Routing Agent has failed over to its standby, the response will be transferred to the newly active instance. Note: There is no response to a [C0].

Apart from time-stamping, routing and eliminating duplicate messages caused by the Riposte network, the Routing Agent will not perform any business or cryptographic functionality. It will relay the [R1], [C0] and [A3] messages transparently. It will not attempt to match an [A3] with the [R1], not time out the expected [A3], nor monitor if the [A3] is received late. The timeout value specified by the Counter in the [R1] is passed to the Authorisation Agent for its use in timing out the response from the FI Authorisation. Note: This timeout technique is that currently used to control the timeout of an [A] response from NBE.

5.1.2.2 Structure, Launch and Concurrency

The NBX Routing Agent has the same structure and analogous characteristics as the NBS Authorisation Agent. As that agent is now obsolescent, much of the information from its design (in [NBSHLD]) is repeated here.

Note that the Routing Agent has much simpler functionality than the NBS Authorisation Agent, which means that the implementation of the Routing Agent has retained characteristics that are not of obvious relevance. For example, it has a pool of [R1]-Worker threads, which now essentially do nothing except copy the [R1] unchanged; it times out [A2] responses, now only for housekeeping purposes, even though the functional processing of late [A2]s is identical to that of timely [A2]s. The Routing Agent can, therefore, be considered as over-engineered.

The Agent runs as an NT Service, and is launched and relaunched by Tivoli. It will be run under its own Service User name (see Table 30).

A separate instance is run against each Riposte Cluster. It will not be dependent on any Correspondence Server in a different Cluster, nor will it be dependent on the Cluster Lookup Service.

A single instance of the Agent will be capable of supporting the entire workload for a Cluster. However, a second instance will run, typically at the other Campus, to act as a hot **Standby** (see 5.1.2.6). In this document, an Agent instance regards the other instance in such a **Resilient Agent Pair** as its **Partner**.

5.1.2.2.1 Threads

Whilst the thread structure of the Agent can be considered as the province of the Low Level Design, it aids this description of the Agent to at least mention it.

In order to handle the required throughput the agent will include at least the following threads for the business functionality.

- A pool of **[R1]-Listener Threads** (also known as **Message Port Threads**) to receive [R1]s and [C0]s and to suppress duplicates
- A pool of **Comms Handler Threads** (also known as **[R2]/[A2] Threads**) to communicate with the NBX Authorisation Agents
- A pool of **[A2]-Worker Threads** to process [A2]s and write corresponding [A3]s

The number of threads in each pool will be configurable (by registry).

- There needs to be precisely one [R1]-Listener Thread for each real-time message port, i.e. one per Agent_Hash value.
- There needs to be precisely one Comms Handler Thread for each target Logical FI (i.e. for each logical NBX Authorisation Agent).

These threads are *persistent*, for the inherent performance benefit.

In addition to these business-related threads, threads are needed for control purposes: in particular, the **Heartbeat Thread**. This thread¹, is responsible for various miscellaneous tasks that are required for controlling the Agent and for monitoring the resources upon which the Agent is dependent. In effect, it manages all aspects of resilience.

- Generating Heartbeat messages indicating the health of the Agent to its Partner
- Monitoring the Heartbeat messages from its Partner, and making decisions on failover
- Monitoring the connection to Riposte, and switching between Correspondence Servers
- Monitoring the health of the communication with the Authorisation Agents, using Application Pings

¹ As will be seen later, there are actually two Heartbeat Threads, with the second monitoring the first, each connecting to a different Correspondence Server.

- Monitoring other necessary resources

The Heartbeat Thread listens on a priority real-time message port for Heartbeat messages from its Partner (ignoring the ones it has generated itself). The rationale is as follows:

- Priority messages overcome the loss of the network path (priority messages are broadcast along multiple paths)
- A real-time message port (of any sort) overcomes any latency problem with Riposte replication. (Note that this means the underlying Heartbeat transport is no longer Riposte replication, rather it's effectively a datagram approach)
- A priority real-time message port ensures that only priority broadcasts are passed to the message port. The Agent does not have to use time-based semantics to distinguish obsolete Heartbeats that have been delayed in replication².

5.1.2.2.2 Heartbeats

The exchange of Heartbeat messages between an Agent and its Partner is used:

- to vote as to which of the two Agents is the **Active Agent** and which the **Standby**
- to control failover from the Active Agent to the Standby, thereby reversing roles
- as a source of status information for operational management of the Agent – the Heartbeats are harvested to OMDB (see 6.2.1.1)
- as a source of data for system management purposes (see 6.2.2.1). To prevent overload, this additional data is not included in every Heartbeat message. Its frequency of inclusion is configurable

An important aspect of monitoring Heartbeat messages from one's Partner is to detect when one or more expected Heartbeats have not been received. The Standby Agent is expecting to receive a Heartbeat from the Active Agent on a regular basis. A slight leeway is allowed before a Heartbeat is deemed to be missing.

This topic is discussed in 5.1.2.6.1.

5.1.2.2.3 Monitoring the connection to Riposte

As discussed in the section on resilience (5.1.2.6.2), in the event of a failure in the connection to Riposte on a Correspondence Server, the Agent attempts to fail over to use an alternative Correspondence Server in the same Cluster.

The Agent uses the Riposte message port in the Heartbeat Thread to monitor the health of its connection to the Correspondence Server. In the event of a failure, it causes [R1]-Listener Threads to switch to the alternative Correspondence Server³. Each [R1]-Listener Thread must start with an empty message port to prevent any possible reprocessing of an [R1] that has already been processed – indeed the message port for the new Correspondence Server will not be created until the switchover.

Even though a short timeout is used when reading from the message port, network-level retries between the NBX Routing Server and the Correspondence Server can cause the call to Riposte to hang for up to 16 seconds. Such a delay could mean that up to three Heartbeats would not be generated, sufficient to trigger the Standby Agent to take over. To avoid this, there is actually a second Heartbeat Thread monitoring the first one, and to provide a fast failover to the alternative Correspondence Server.

² It significantly strengthens the protocol to avoid making any assumptions about clock synchronisation between Nodes.

³ There is actually a reserve pool of [R1]-Listener Threads waiting to be activated, the details of which are beyond the scope of this HLD.

5.1.2.2.4 Application Pings – Monitoring the connection to an Authorisation Agent

At times, it is necessary to check that an NBX Authorisation Agent is available. A ‘Ping’ status message pair has been defined to achieve this. Their formats are defined in 5.4.1. A new socket is created for each Ping; it is disconnected after use.

Pings are sent by the Agent under the following circumstances:

- When establishing the initial connection
- Routine monitoring during periods of inactivity (such as overnight). The configurable interval is typically of the order of several minutes (maybe 10 minutes).
- A potential problem has been flagged by a Comms Handler Thread. The Heartbeat Thread will send the Ping to determine whether the potential problem is real. Possible reasons include:
 - Lost socket connection
 - Unexpected (typically late) [A2]
 - Timeout on receiving [A2]
- “Emergency” Ping by the Standby Agent following receipt of a Heartbeat message from the Active Agent indicating that it may be having problems. This Ping is used as part of the decision-making whether to fail over to the Standby Agent.
- Monitoring whether an NBX Authorisation Agent is still Down. Periodic retries are made at configurable intervals.

The Heartbeat Thread imposes a minimum interval (configurable) between successive Pings for a particular Authorisation Agent. This is to avoid sending a separate Ping for each [A2] affected by the same potential problem. The one exception to this is that the Emergency Ping used during failover voting is always sent immediately.

Having sent the Ping, a corresponding response is expected in return. The NBX Authorisation Agent is considered to be “**Down**” if the response has not been received within a configurable timeout (of the order of 30 seconds, longer than the network self-reconfiguration time). A Heartbeat message will inform its Partner of the failure.

5.1.2.2.5 Monitoring other necessary resources

If the Agent is unable to offer a routing service because, for example, it is unable to access a critical resource, the Agent will deem itself to be “**Unavailable**”. A Heartbeat message will inform its Partner of its unavailability. *(There may be no resource that the Agent deems to be this critical.)*

The Agent is also Unavailable if it is closing down, for whatever reason. Failover is quicker and cleaner if this fact can be passed to the Agent’s Partner, but it may not be possible in all failure scenarios.

5.1.2.2.6 Inability to offer a routing service

There are two scenarios where the Agent is unable to offer a routing service, or, in the first scenario, a service for only certain target Authorisation Agents:

- The Routing Agent has detected operationally that an Authorisation Agent has failed (i.e. is **Down**)
- The Agent itself is **Unavailable**

On entering any of these states, the Agent will close all sockets connecting to affected Authorisation Agents, and wait long enough (perhaps one minute, configurable) to be sure that the underlying TCP/IP disconnection has completed.

If the problem results in a controlled failover to its Partner, the Agent will discard all the [R1]s and [C0]s that it has not yet forwarded (see 5.1.2.6.1.2).

5.1.2.3 Functional Description

5.1.2.3.1 Real-time message ports

The active Routing Agent (but not the standby) listens on a priority real-time message port to pick up all banking [R1] and [C0] messages. A priority real-time message port is used for two reasons.

- It ensures the minimum delay before a message is processed following its arrival at the Campus
- Any [R1]s that initially fail to be transmitted will be ignored when they are subsequently replicated to the Campus. Such messages will be 'stale', and there is no point in processing them. (There is a separate agent, the NBX Guaranteed Reversals Agent, to harvest any [C0]s that may have been missed by the Routing Agent.)

A separate real-time message port is used for each value of the **Agent_Hash**, <AgtHash>, a value that ranges from 0 to 3. The Agent_Hash is algorithmically derived from the Branch's FAD Code at the Counter and is included in the [R1] and [C0] messages.

A real-time message port will generally present each Riposte message up to seven times to the Agent in the current Horizon configuration. This is a consequence of the complex network – including multiple IP addresses on multiple LANs – fronting the Correspondence Servers. The Agent uses the **Horizon_Txn_Num**, <HTxnNum>, and the message type (R1 or C0) to identify the message.

The filter on the priority real-time message port used to read the [R1] and [C0] messages is:

- <Application:NBA>
- <Data.Ctrl.MsgType:R1> or <Data.Ctrl.MsgType:C0>

Note that the Routing_Gateway value is not used as part of this filter. The Routing Agent accepts all banking [R1]s and [C0]s and attempts to route them (see below).

Duplicate [R1]s (or [C0]s) are discarded. The check is performed initially by inspecting a cache local to the thread handling the message port, and definitively when an entry for the [R1] (or [C0]) message is inserted into an Agent-wide, in-memory table of "Requests being processed", this table's main purpose being to detect duplicate messages. Messages will be retained in this table for a minimum of 15 seconds (configurable by registry) to allow for the late arrival of duplicates.

The time the message was read from the message port is retained for passing along with the message to the Authorisation Agent. This is termed the **Routing Agent timestamp**.

5.1.2.3.2 Routing the [R1]/[C0]

The main function of the NBX Routing Agent is to route [R1]s and [C0]s to the NBX Authorisation Agent servicing the target Logical FI (e.g. CAPO_A).

The configuration data driving the routing algorithm operates at two levels:

- NBX Routing Data: This Type D Reference Data is used to identify the target Logical FI – see 5.1.1.4.1 and 5.5.1.2
- List of target Logical FIs: This registry parameter is used to configure one comms handler thread for each of the Logical FIs.

The Routing Agent reads the Routing Data during its initialisation and caches it in memory – the cache is never refreshed. Whilst caching it, the Agent performs consistency checks on this configuration data, and terminates with an Error event if there are inconsistencies. These checks include:

- The Routing Data identifies at least one target.
- Every target in the Routing Data exists in the (registry) list of target Logical FIs.

- The Routing Data is complete, i.e. every combination of Cluster number and Agent_Hash is covered for any Routing_Gateway value.
- The Routing Data is not ambiguous, i.e. any particular set of routing criteria do not result in two or more conflicting targets.

If a message cannot be routed because there is no matching Routing Data, the behaviour depends upon the state of a registry switch. During the period of migration, when the NBX system is only processing specific Routing Gateway values, the NBX Routing Agent will silently ignore the message and leave it to NBE to process appropriately. Once the NBE has been switched off, the registry switch will be thrown and the unroutable message will be treated as an exception. It will be still be ignored, but it will be in the NT Event Log. Measures will need to be taken to prevent an event storm in the event of a large number of exceptions.

A successfully routed message is then queued for the comms handler thread for the target Logical FI.

5.1.2.3.3 Comms Handler Threads

The Comms Handler Threads provide the interface to the NBX Authorisation Agents. There is a separate thread per (logical) target NBX Authorisation Agent.

Each Comms Handler Thread in an active Routing Agent establishes and maintains connections to both the active and standby Authorisation Agent instances – see 5.1.1.3.1 for a discussion of the topology between the Routing Agents and the Authorisation Agents.

A Comms Handler Thread may be configured to establish more than one socket to communicate with an Authorisation Agent instance, but as stated earlier the default configuration is for just one.

5.1.2.3.3.1 *Configuration of Host and Service names*

The Comms Handler Threads are configured (collectively via registry) with the set of host and service names for the ports on which the Authorisation Agents are listening. The registry values for these names contain substitution parameters, substituted at run-time (not build-time):

- %T for target Logical FI
- %C for Cluster number
- %V for virtual address index, which is 1 for Authorisation Agents at Bootle and 2 for Wigan

The names, thus expanded, are resolved in the Hosts file into an IP address and in the Services file into a port number. The proposed registry values are:

- SOCKET_HOST = TMSNX_%T_VA%V
- SOCKET_SERVICE = TMSNX_%T_CL%C

Thus the Hosts file would need to contain, for example, TMSNX_CAPO_A_VA1 and TMSNX_CAPO_A_VA2 for the Routing Agents to connect to the Authorisation Agents for CAPO_A at both Bootle and Wigan. Similarly, the Services file would need to contain TMSNX_CAPO_A_CL1 for the Routing Agent for Cluster 1 to connect to the Authorisation Agents for CAPO_A at both Bootle and Wigan.

5.1.2.3.3.2 *Inter-Agent protocol*

The message protocol between a Routing Agent and an Authorisation Agent is given in 5.4.1.

As described there and in 5.1.1.3.1, before any banking messages are transferred on such a connection, there is an exchange of status messages, whereby each of the Routing and Authorisation Agents declares whether it is active or standby. Furthermore, each Agent will send another status message whenever its status changes (from active to standby or *vice*

versa). By this means, a Routing Agent knows which Authorisation Agents are active – it will send banking messages only to an active Authorisation Agent.

If the Authorisation Agent has indicated by means of a HOLD flag in an ASTS status message, the Routing Agent must refrain from writing any banking messages to that socket. This suspension will persist until countermanded by an ISOK flag on a subsequent ASTS.

When an [A3] (or exceptionally an [A4]) response is read from a socket, it is processed regardless of whether the Agent is active or standby. Processing consists of writing it to Riposte as a normal (i.e. non-Priority) message. If practicable, it will be given the same expiry period as the corresponding [R1].

For the [A3] to be immediately replicated from the Correspondence Server to the Counter, it requires that the network connection has been kept open. When the Counter wrote the [R1] as a priority message, it will have set the Riposte connection parameters to ensure this.

For historic reasons (the development of the NBX Routing Agent from the NBS Authorisation Agent), this thread will attempt to match responses against a list of outstanding responses. A (longish, perhaps 30 or 40 seconds) timeout value will be applied. But both these actions are purely for housekeeping purposes and in no way affect the business functionality. A timeout will be regarded as a possible problem with the Authorisation Agent, and will be flagged to the Heartbeat Thread accordingly (see 5.1.2.2.4).

5.1.2.3.3 *Other considerations*

A Comms Handler Thread has a number of tasks to schedule. Non-blocking writes and reads are used. If there is no immediate work to do but there are outstanding responses, it sleeps for a short while (20 milliseconds, configurable); if there are no outstanding responses, it sleeps for a longer while (200 milliseconds, configurable).

The messages to be written are distributed amongst the sockets on an essentially round-robin basis, whilst ensuring that the load does not get too unbalanced. This balancing caters for the possibility that a socket can get irrecoverably “frozen”, e.g. its TCP window has been reduced to one and its throughput is much reduced.

The Thread has to balance writes against reads. The details are beyond the scope of the HLD, but essentially reads take priority when an Authorisation Agent already has been given many messages to process, and writes takes priority when it has few.

If the Authorisation Agent has stopped responding – if no expected responses have been received for 5 seconds (configurable) – it is desirable to flag this as a possible problem immediately. The Heartbeat Thread can then initiate the Application Ping to probe whether the Authorisation Agent is Down, without waiting for a specific response to be timed out – this could lead to a saving of 10 seconds in the failover process.

Problems such as a lost socket connection are regarded as possible problems with the Authorisation Agent, and are also flagged to the Heartbeat Thread.

5.1.2.4 **Exception Handling**

Operational failures are treated by this Agent in a different way from traditional Harvester Agents. This is partly because of the requirements of high availability, which has led to the concept of a Resilient Agent Pair and of failover to a Standby Agent (see the section on Resilience, 5.1.2.6).

Note that all failures and exceptions are recorded in the NT event log.

5.1.2.5 **Performance and Scalability**

5.1.2.5.1 **Targets**

[BUSVOLS] provides NBX business volumes for the Routing Agent. The volumes are expressed as the number of transactions per second for the peak five-minute period of the

month. The **Contracted Volume** figure is the maximum volume that Fujitsu Services will contract to support. The **Design Limit** figure is the volume that the system will support without significant failures. These targets have been further analysed in Table 2 for the four Routing Agent instances, one per Cluster.

Item	Contracted Volume	Design Limit	Comment
Average NBX transaction rate in peak 5-minute period across all Clusters	204	245	Total volumes from [BUSVOLS]
Average transaction rate in the worst Cluster	58	72	Cluster 1 is expected to be 28% of the total load
Target transaction rate per Agent in the worst Cluster	67	83	Target is that 99.5% of transactions queue for < 0.5 seconds

Table 2 – Routing Agent Performance Targets (transactions/sec)

Note that these are the number of [R1] messages to be processed, to which must be added the Reversals. The normal proportion of Reversals is low, maybe 2%, but under periods of overload where stale [R1]s have to be discarded, the number of Reversals can be as many again.

5.1.2.5.2 Analysis

There are no mill-intensive operations to be performed, so the number of worker threads is not critical to achieve the throughput.

5.1.2.5.1.1 *Agent_Hash values for Riposte message ports*

A Riposte real-time message port accessed remotely over RPC was originally said to be capable of delivering one message every 5 ms, equivalent to a throughput of 200 messages/sec. This is the figure used in the original calculation, but which James Stinchcombe now says is exceptionally pessimistic – a throughput of 800 messages/sec is now the correct figure to use.

As stated in 5.1.2.3.1, each [R1] can be delivered seven times. Therefore, the (pessimistic) throughput achievable through a single message port is 30 transactions/sec.

It was therefore decided to continue the approach used with similar agents, with one message port per Agent_Hash value. Four Agent_Hash values, as traditionally used for the Network Banking Service, are more than sufficient for this purpose. It requires the transactions to be reasonably well distributed across the Agent_Hash values, which has been shown to be the case.

For the Routing Agent, the four message ports are to be handled by having four [R1]-Listener Threads (aka Message Port Threads) in a single Routing Agent instance.

5.1.2.5.1.2 *Connections to the NBX Authorisation Agents*

A Comms Handler Thread may be configured to establish more than one socket to communicate with an Authorisation Agent instance, but as stated earlier the default configuration is for just one. This is quite sufficient to achieve the throughput.

5.1.2.6 Resilience

Mechanisms are required to ensure that, should an instance of the Agent fail, another one will take its place. The case of a Correspondence Server failure (including a failure of the network linking the Agent Server to the Correspondence Server) also needs to be considered. In this case, the Agent process will continue operating, but use the other local Correspondence Server.

In the case that an Agent instance fails, then all outstanding transactions being processed by that Agent will be abandoned (and will eventually timeout at the Counter). No attempt will be made by the replacement Agent to recover such work.

5.1.2.6.1 Agent Failover

Resilience to Agent failure is achieved by having two instances of the Agent running at the same time on different platforms (typically at different Campuses), one running as the active Agent, the other running as a standby. When an Agent instance first starts up, it runs as a standby.

If the active Agent fails, the Standby Agent takes over and become the active one. For this approach to be viable, a reliable communications link is required between each instance of the Agent. This is provided by the Riposte service running on the Correspondence Servers, which provides reliable communications between nodes within a Riposte Cluster. A key feature is the ability to use multiple network connections to communicate to other nodes within the Cluster. Failover to another network connection occurs within 10 seconds of the failure being detected.

5.1.2.6.1.1 Heartbeat messages

Each Agent instance will write a Heartbeat message to the Riposte message store every five seconds, except for the standby Agent when there is a fully functioning active Agent, when it is every 15 seconds (both configurable).

The Heartbeat message will include the following information for failover purposes (there may be additional data for systems management purposes):

- *Active* – Whether the Agent instance is actively processing banking transactions. Values of ‘Y’ and ‘N’ will be used
- *Capabilities* – Whether the Agent instance is fully capable of offering a routing service. It comprises separate indicators:
 - *Unavailable* – Whether the Agent instance can offer a routing service at all (see 5.1.2.2.6). Values of ‘Y’ and ‘N’ will be used
 - *Authorisation Agent Problem* – The status of the Agent instance’s connection to the Authorisation Agents collectively. Values of ‘Y’ and ‘N’ (meaning Down and Up respectively) will be used, the value will be determined using Application Ping mechanism described in 5.1.2.2.4. If only some Authorisation Agents are Down, then a composite value is used, e.g. ‘YYNYYY’, in the same order as the target Authorisation Agents are declared in the registry.

The composite value is there for harvesting to OMDB. For the rules below, the ‘partially Down’ value is equivalent to Down.

- *Authorisation Agents Confirmed OK* – In a controlled failover (see below), confirmation that the status of the Agent’s connections to the various (active) Authorisation Agents is up-to-date, having just been checked by an Emergency Ping. Values of ‘Y’ and ‘N’ will be used
- *Unilateral Takeover Imminent* – A warning that the Agent will unilaterally take over following the next missing Heartbeat (see below)

- *Priority* – A static priority value obtained from the NBX Routing Server registry. This is used to determine which Agent has priority at Agent service start-up. It is also used to determine which Agent should ‘resign’ when they are both active – this can happen following a break in the inter-Campus link. The Values 1 & 2 will be used, with 1 being the higher priority
- *Sequence Number* – Assigned to each Heartbeat so that the receiving Agent can tell whether it has arrived out of sequence and duly ignore it. This can arise because its Partner can exceptionally write Heartbeats to two different Correspondence Servers, each with different delays. It is also the mechanism used to ignore duplicate Heartbeats received because of the use of a real-time message port. The sequence number starts at one when an Agent instance is launched – to allow for this, the receiving Agent will accept an out-of-sequence Heartbeat following a missed Heartbeat

As Heartbeat messages are also used for operational monitoring of the Routing Agent instances, and are therefore harvested to OMDB, an Agent instance will write a Heartbeat even if it is operationally incapable. This is so that information is available to aid diagnosis of the problem.

Heartbeat messages will be written to a reserved group, 999991, within the message store to which it is connected. This group, called the **Heartbeat Office**, will exist in all Clusters. Two instances writing Heartbeat messages of 225 bytes, or 375 bytes with systems management data, will result in 70 Mb⁴ of uncompressed space being required per Cluster. With data compression, it is expected that this space will reduce significantly.

Each Agent instance will connect to both Correspondence Servers in its Cluster at the local Campus, so that it can receive Heartbeat messages from both. (Note that they are written to only one Correspondence Server.) Failure of either connection will require the Agent to enter a retry loop until connection can be re-established. Failure of both connections will result in the Agent exiting, at which point it will be restarted by EACRR.

Each Agent instance will attempt to write a final Heartbeat message when it closes down, in both normal and exception scenarios, to inform its Partner what it is doing. *Active* will be set to ‘N’ and *Unavailable* to ‘Y’. (PC/78184)

There are two distinct failover scenarios to be considered:

- Both Agent instances are running and can control the decision
- Expected Heartbeats from the active Agent are missing

The Heartbeat message is of the generic format first defined for the ETS Authorisation Agent (see [ETSHLD] and [OMDB]), with the NBX Authorisation Agents collectively being the “Enquiry Engine”. The <Application:> and <Data.Heartbeat> attributes will both be set to “NXRTNG”.

5.1.2.6.1.2 *Controlled failover*

There are two aspects to a controlled failover. Firstly, upon receiving a Heartbeat, an Agent instance has to decide whether it is the one that should be the active Agent. Secondly, the transfer of control has to be managed.

The decision-making is as follows:

- *Relative capabilities* – The Agent instance compares its current capabilities with those of its Partner’s heartbeat
- *Active* – If the capabilities are equal, an active Agent takes precedence over a standby Agent

⁴ Calculation assumes that the Active Agent writes two short and one long Heartbeat message, and that the Standby Agent writes one short Heartbeat message, in every 15-second period.

- *Priority* – Everything else being equal, the Agent with the lower priority number will take precedence

The essence of the failover protocol is as follows. This considers the ‘normal’ case where the active Agent has deemed the NBX Authorisation Agents to be Down.

- The active Agent’s Heartbeat indicates an Authorisation Agent Problem
- Upon receiving this indication, its Partner checks its own Authorisation Agent capability by doing an immediate Emergency Ping of the Authorisation Agents (see 5.1.2.2.4). If it is OK, it writes a Heartbeat recording its superior capabilities and with an “Authorisation Agents Confirmed OK” flag set
- Upon receiving this Heartbeat, the active Agent’s decision-making realises that its Partner is the more capable and has just confirmed that its capability information is up-to-date. It unilaterally relinquishes being the Active Agent, and writes a Heartbeat saying that it is not Active
- Upon receiving this Heartbeat from a now inactive Agent, the standby Agent makes itself the active Agent and writes a Heartbeat to proclaim this

When an active Agent relinquishes being Active, it discards all the banking transactions that it has partially processed, letting the Counters time them out. In addition, to avoid any danger of rapid ‘ping-ponging’ during transient failures, the Agent will not write a Heartbeat with the “Authorisation Agents Confirmed OK” flag set for at least three minutes (configurable).

5.1.2.6.1.3 Unilateral takeover

A standby Agent instance will wait for the non-receipt of Heartbeat messages for three Heartbeat periods (configurable) before unilaterally taking over as the active Agent and processing NBS transactions. A Heartbeat is not deemed to be “missing” until after a slight leeway (1 second, configurable) has elapsed after the time it was expected. (*Note that NBX Routing Agents do not use the concept of “pre-emptive mode” that the NBS Authorisation Agent used.*)

The final decision to take over has to guard against delays in replicating the Heartbeats between the Correspondence Servers (typically across the inter-Campus link). Because of this, on the missed Heartbeat before the final one the Standby Agent, in its Heartbeat, warns its Partner that “takeover is imminent”. When (and if) the Partner receives this warning, it delays any decision that it may have been about to take.

(Note: If the next Heartbeat received after one has gone missing has a lower Sequence Number than expected, the “expected Sequence Number” is reset – this is to allow for the Heartbeat having been generated by a replacement instance of the Partner.)

5.1.2.6.1.4 Duplicate messages

The resilience strategy has tried to avoid passing the same [R1] or [C0] message to an NBX Authorisation Agent more than once. However, there may be exceptional conditions, for example following the reestablishment of a failed inter-Campus link, that both Routing Agent instances in a Resilient Agent Pair do forward the same message, possibly even to different instances of the one ‘logical’ NBX Authorisation Agent.

This does not matter. As an NBX Authorisation Agent records every transaction in the NBX Status File, the NBX Authorisation Agent will notice the duplicate [R1] or [C0] and will not process it, beyond logging a duplicate [R1] (but not a duplicate [C0]) in the NBX Transaction Journal.

5 Priorities will only be equal in a misconfigured system. To allow for this, the Agents will use a final discriminator, such as alphabetical order of host name.

6 This delayed decision is material only when neither Agent is Active, as can happen during start-up, and is to prevent both Agents becoming Active.

5.1.2.6.2 Correspondence Server Failure

There are two Correspondence Servers per Cluster at each Campus. Each NBX Routing Server will be physically connected to both Correspondence Servers, using separate network cards and separate LANs, such that an NBX Routing Agent should at all times be capable of connecting to one or other of them.

When a Correspondence Server fails or is removed for routine maintenance, any NBX Routing Agents connecting to the Correspondence Server need to automatically fail over to use the other Correspondence Server. Failover should be achieved within a few seconds, therefore minimising the period that Outlets serviced by the Cluster are affected.

This failover will be achieved within the Agent application, as described in 5.1.2.2.3. The configuration of the Agent with a **Resilient Locale** is described in 8.1.1. Once it has switched Correspondence Servers, it will try to re-establish

Should the Agent lose its connection to both Correspondence Servers, it will fail so that EACRR may restart it. To achieve a restart within about one minute, EACRR will be configured to restart it on the same NBX Routing Server rather than on another one.

During the connection phase when the Agent is first loaded, the Agent attempts to connect to both Correspondence Servers. Until it has succeeded connecting to at least one of them, it adopts the standard Agent approach of retries as appropriate; retries continue until the configured TOTAL_CONNECTION_TIMEOUT period has elapsed – this will be set to 5 minutes.

Once it has connected to one Correspondence Server, the Agent can enter its main processing phase. Attempts to connect to the other Correspondence Server continue indefinitely (and are not controlled by the TOTAL_CONNECTION_TIMEOUT).

Similarly, following a failover from one Correspondence Server to another, attempts to re-establish resilience are made, by trying indefinitely to reconnect to the original Correspondence Server.

5.1.2.6.3 Network Failure

The Mean Time Between Failure (MTBF) for a LAN is generally accepted to be in the region of three weeks. This means that network failures will be the most likely cause of Agent problems. The Agent connects to two resources using the network – the NBX Authorisation Agents and the Correspondence Server.


Failure of the network route to a Correspondence Server is treated in the same way as failure of the Correspondence Server itself.

5.1.2.6.4 NBX Authorisation Agent failure

An NBX Routing Agent will not fail on failure of the connection to an NBX Authorisation Agent.

5.1.2.7 Configurability

Table 3 lists the more important items that are configurable through the Registry.

 The default values given in this table are indicative only. The reader should consult the Low Level Design document [NBXROUTING] for authoritative default values and names of the Registry values.

Item	Description	Section	Default
------	-------------	---------	---------

Target Logical FIs	List of the Logical FIs serviced by each of the NBX Authorisation Agents, e.g. "CAPO_A,CAPO_B,LINK_A,LINK_B,AL_A,AL_B". These names must match those in the NBX Routing Data The associated count gives the number of Comms Handler Threads, with one thread per Logical FI	5.1.2.2.1	–
NBX Authorisation Agents' host names (SOCKET_HOST)	Used to look up the IP addresses. The host names must follow a particular naming convention	5.1.2.3.3.1	–
NBX Authorisation Agents' service names (SOCKET_SERVICE)	Used to look up the port numbers. The service names must follow a particular naming convention	5.1.2.3.3.1	–
Number of virtual addresses per Comms Handler Thread (VA_COUNT)	One for each of the NBX Authorisation Agents in a Resilient Agent Pair	5.1.2.3.3	2
Number of sockets per Comms Handler Thread (SocketConcurrency)	One for each of the NBX Authorisation Agents in a Resilient Agent Pair. Excludes the one used for Pings	5.1.2.3.3	2
Allowance before replacing closed socket	Minimum delay before attempting to create a replacement socket following the loss of a socket		2 secs
Resilient Locale	Name of a Resilient Locale configured in the CLUSTER_LOOKUP_SERVER registry	8.1.1	
Cluster number (CLUSTERID)	Number of the Riposte Cluster for this Agent instance		
Range of Agent_Hash values	It is possible to configure an Agent instance to process a subset of the Agent_Hash values		All (i.e. 0–3)
Minimum hold time for duplicate checking	Minimum time an [R1] is held in a memory table for the suppression of duplicates delivered by the priority real-time message port	5.1.2.3.1	15 secs
Minimum period between Pings	Minimum interval between completing one Ping and starting the next	5.1.2.2.4	30 secs
Check period when idle	Period without receiving messages from an NBX Authorisation Agent. If none received, a Ping is initiated	5.1.2.2.4	10 mins
Ping timeout	Time after which an NBX Authorisation Agent is deemed to be Down	5.1.2.2.4	30 secs
Late response	If no expected response messages are received from an NBX Authorisation Agent for this interval, a potential problem is flagged	5.1.2.3.3.3	5 secs

Quick sleep	Duration of short sleep when Comms Handler has nothing to send but responses are expected	5.1.2.3.3.3	20 msec
Wait sleep	Duration of short sleep when Comms Handler has nothing to send and no responses are outstanding	5.1.2.3.3.3	200 msec
Heartbeat interval (normal)	Used except when the following entry applies	5.1.2.2.2 5.1.2.6.1.1	5 secs
Heartbeat interval (slow)	Used by standby Agent when the active Agent is functioning fully	5.1.2.6.1.1	15 secs
Heartbeat leeway	Allowance before an expected Heartbeat is deemed to be missing	5.1.2.2.2	1 sec
Missing Heartbeats for takeover	Number of missing Heartbeats, after which the standby Agent unilaterally becomes Active	5.1.2.6.1.1	3
Controlled takeover delay	Minimum interval after an active Agent has stood down before it will become Active again by controlled takeover	5.1.2.6.1.2	3 mins
Priority	Different priorities need to be assigned to each Agent instance	5.1.2.6.1.1	1, 2
Heartbeat Office	Group Id used for Heartbeat messages	5.1.2.6.1.1	999991
Heartbeat system-management interval	Frequency of including system management data in Heartbeats	5.1.2.2.2	15 mins
Agent Ref Data Office	Base Group Id used for NBX Routing Data. Actual Group Id used is this base value plus the Cluster number	5.5.1.2	999960
Exception Rules	Control of how to handle exceptions. During migration, the default will be to ignore unroutable messages	5.1.2.3.2	

Table 3 – Configuration of NBX Routing Agent

5.1.2.8 **Audit**

As the NBX Routing Agent is not servicing an external interface there is no requirement for auditing the messages being processed.

5.1.2.9 **Operational Summary**

Agent name: NX_NQ_RTNG	Platform(s): NBX Routing Server
Service Name: TMSNXRtng0_<s><n>	Style: Resilient Enquiry Agent, with Hot Standby
Scope & parallelism: One per Site per Cluster (<s> is the site, <n> is the Cluster id); multithreaded. One in each Cluster is acting as the Hot Standby for the other.	
Registry key(s): HKEY_LOCAL_MACHINE\SOFTWARE\ICL\PathwayAgents\NX_NQ_RTNG HKEY_LOCAL_MACHINE\SOFTWARE\ICL\PathwayAgents\NX_NQ_RTNG\TMSNXRtng0_<s><n>	

Use of checkpoints: None.
Use of dummy offices: The Heartbeat Office (999991) for Heartbeat messages; this dummy office must exist in all Clusters. Agent Reference Data Office (99996< <i>n</i> >), one per Cluster.
Host database: None. Dependent instead on the NBX Authorisation Agents.
Needs to be running: Runs 7 x 24. Important between 07:00 and 20:00 7 days per week; Critical between 08:00 and 17:30 Monday to Friday and 08:00 and 13:00 Saturday.
Documentation: [NBXROUTING]

5.1.3 NBX Authorisation Agents (NX_NQ_CAPO, ..._LINK, ..._AL)

The Low Level Design document for this Agent is [NBXAUTH].

5.1.3.1 Business Functionality

An **NBX Authorisation Agent** is responsible for handling all [R1] and [C0] messages received from an NBX Routing Agent, and the corresponding [A1] and [E2] response messages returned from an FI's Enquiry Engine (FI_EE).

The NBX Authorisation Agents for different FIs are separate executables, so that they can be developed, configured, maintained and operated independently of each other, though they will naturally share much source code. They will be distinctly compiled executables.

As required by the Technical Interface Specifications, there will be two 'logical' NBX Authorisation Agents for each FI. For CAPO this is primarily to meet the volume throughput, for the others for reasons of resilience. Each of the two logical instances will each handle approximately half the estate. The Routing Agent will ensure that all the transactions for a particular Branch, including the [C0] Reversals, are routed to the same logical Authorisation Agent.

Each FI_EE is fronted by two or more Processor Interfaces (PIs). Each PI has a conceptual counterpart within an Authorisation Agent, such that there is a one-to-one mapping between Agent PI Handlers and FI_EE PIs. Business rules state:

- Logon and Logoff commands operate at the PI level.
- Acquirer Working Keys (AWKs) operate at the PI level.
- Responses to messages are returned from the PI that received the request message (but see below).
- [E1] Reversal messages will be transferred to the same PI as the original [R1] request message. Note that for CAPO, the business rule in the TIS is even more stringent and requires that [E1] Reversal messages are transferred to the same remote Thread as the original [R1] message.

Each PI Handler to PI session is represented by one or more TCP/IP connections: currently one for A&L and LINK and two for CAPO. The several connections may either be all to the same remote virtual address (IP address and port) or all to different addresses. (*The Agent has a limitation that it does not cater for several connections to each of several VAs.*) These relationships are fully configurable via registry.

Whether the response to any message sent to the FI_EE is returned over the same TCP/IP connection depends upon the particular FI – each of the three FIs will guarantee that it is.

An NBX Authorisation Agent's primary function is to take the incoming [R1] request, validate it (including the Digital Signature, but see below), reformat it as an [R3] (including any decryption and re-encryption of data), and pass it through to the FI_EE. It then picks up any [A1] Authorisation returned from the FI_EE, validates it, reformats it as an [A3], adds a Digital Signature, and passes it back to the originating Counter.

It provides a timeout mechanism (using a timeout value supplied by the Counter in the [R1]) to ensure that should the FI_EE not return an [A1] in time, the Authorisation Agent will generate an [A3] indicating that this has happened. This is necessary in order to ensure that such delayed FI_EE messages are not recorded as network failures against Fujitsu Services's SLAs. Similarly, an appropriate [A3] message will be returned for any [R1] message that fails during processing (for example in signature checking).

The Agent's other primary function is to process Counter-generated [C0] Reversal messages and pass them through to the FI_EE as [E1] messages to the FI_EE. The Authorisation Agent may also generate other Reversal requests internally – in practice, only when an [A1] is received after its timeout period.

Reversals are 'must-deliver' messages. As well as processing [C0] Reversals received by the direct route from the Routing Agent, it also processes them received by the assured route provided by the NBX Guaranteed Reversals Agent.

The status of a transaction is recorded as a Status Entry in the Transaction Status table in the NBX Persistent Store (NPS) and is held there for 5 days. *(Note that the Status Entry holds only the transaction's current status; it does not hold any history.)* It is updated for every transaction part (e.g. [A1]) or event (e.g. [A1] timeout) for this message. This table provides the assured 'store-and-forward' queuing mechanism needed to ensure repeat attempts are made to forward the [E1] until an [E2] response is received from the FI_EE.

These primary functions are amplified in the next sections of this Overview.

5.1.3.1.1 Processing an [R1] Request

The Routing_Gateway value has already been used to route the [R1] to the correct logical NBX Authorisation Agent.

Before the [R1] is processed, its Digital Signature may be checked. If it fails the check, an [A3] Decline will be generated with the 'Signature Failure on [R1]' response code (see Table 4). It was originally intended to check the Signature on every [R1]. However, throughput testing, as well as analysis (see 5.1.3.7.2), has shown that under heavy load a serious backlog can build up waiting for the single-threaded verification resource (the Layer 7 DLL is necessarily single-threaded). Therefore it was decided, via CP3896, to suppress the checking of Digital Signatures for transactions already protected by having a (single) PIN block present. Note that Change PIN transactions, with two PIN blocks, and Deposit transactions, with no PIN block, are always checked. Suppression is controlled by configuration via registry, and currently is set only for CAPO. Furthermore, to prevent spoofing, a check is suppressed only if the Group and Node Ids in the Horizon_Txn_Id match those in the [R1]'s Riposte red tape.

A staleness check will be performed, based on when the Routing Agent receives and time-stamps the message. If the interval is more than 15 seconds (configurable by the StaleR1.Threshold business parameter), the stale [R1] will not be forwarded to the FI and will not be logged. *(It was originally intended to log stale [R1]s, but once throughput testing showed that the NPS could be a bottleneck under certain load conditions it was necessary to dispense with this logging to ensure the stability of the Agent under overload.)* No [A3] will be generated as the Counter will already have timed out the request. This check is performed twice, once before verifying the Digital Signature, which allows for speedy handling of a backlog, and once immediately before generating the [R3], to handle the case where there has been a delay when, initially, there is no available PI.

A duplicate [R1] will be detected by checking against the Transaction Status table in the NPS (see 5.1.3.1.5)⁷. A duplicate could arise in very exceptional conditions with the NBX Routing Agent and its standby. No further action is taken: the duplicate [R1] will be logged and will not be allowed to reach the FI. The Agent is optimised so that there will be no unnecessary read of the Transaction Status table in the normal case where there is no duplicate.

In the unlikely event that the [R1] is processed after the [C0] that reverses it, the [R1] will not be forwarded to the FI but will be logged. The [C0] will not be processed further.

The Agent will dynamically select a PI and a particular TCP/IP connection for that PI. The selection across the available connections will be on a round-robin basis. If all connections are available, the Agent will ensure a different PI is selected for each successive [R3], though this algorithm will be modified for PIs not logged on and for failed or otherwise unusable connections. The Agent is multithreaded and processes messages in parallel, so at busy

⁷ The key used in the Transaction Status table to identify the transaction is the transaction's Receipt Date (expressed as YDDD), the FAD Code and Node Id of the Counter (expressed as a Terminal Id), and the last 6 digits of the fourth component of the transaction's Horizon Transaction Id (guaranteed unique within a day at the Counter, as (a) the Counter commits to Riposte after writing the [R1], and (b) a banking transaction cannot be transferred between Counters).

periods the round-robin may only be observable statistically. More details of load balancing are given in 5.1.3.2.4.

All [R3] messages as well as all [E1] Reversal messages will be treated with equal priority.

If no PI is available, an [A3] Decline will be generated with the appropriate response code. The first time this happens, the Agent will delay processing the [R1] for up to 18 seconds (configurable by the `Timeout.WaitToSend` business parameter) in the expectation that a PI will become available.

The [R3] message is constructed according to the Mapping Documents: [CAPO_MAP], [LINK_MAP] and [A&L_MAP]. The PIN Block is encrypted with the current AWK for the chosen PI. Much of the mapping is controlled by 'soft' configuration parameters – see NBX Business Parameters [BUSPARAMS]. This is particularly true where the mapping is in the form of a 'look-up', such as the message types and processing codes for different Transaction Types, the Point of Service Entry Mode, and the cardholder authentication method in Point of Service Data. On the other hand, the PIN and ICC data fields are derived dynamically from the presence of that data in the [R1] message.

If the Transaction Type is such that the mapping rules are unable to generate an [R3], an [A3] Decline will be generated with an 'Invalid [R1]' response code (see Table 4). This could happen, for example, if faulty Reference Data at a Counter caused a Withdraw Limit Transaction for LINK.

5.1.3.1.2 Processing an [A1] Authorisation

If the Agent times out the [A1] response, it will generate an [A3] Decline with an 'FI Timeout' response code (see Table 4).

When an [A1] is received, the Agent will attempt to match it against the corresponding [R3] request. The match will be performed, in the first instance, against in-memory tables of recent [R3]'s, failing that against the Transaction Status table in the NPS. If it cannot be matched, then it will not be possible to reverse it (if it were approved) as there is insufficient information in the [A1] alone to generate the [E1].

If an [A1] Approve is received after the timeout period, the Agent will generate an [E1] Reversal for sending to the FI. If an [A1] Decline is received under similar circumstances, no Reversal will be generated.

The Agent will construct an [A3] for a timely, matched [A1] according to the mapping rules, and will return it to the (logical) Routing Agent from which it received the [R1].

The [A3] includes a Digital Signature. If it cannot be signed due to a further failure, then an unsigned [A4] is written in its stead. An [A4] is the same as an [A3] except that it has a different message type and no Digital Signature. It is generated purely for diagnostic purposes, as the Counter will not process it.

If a local error, such as the loss of the connection to the Routing Agent, prevents the Agent from returning an [A3], the Authorisation Agent will hold the [A3] until a connection is re-established. It will not itself generate an [E1] Reversal. Instead, it will let the Counter time out the [A3] and generate a [C0] Reversal, which will be processed in due course.

A Failure-[A3] will be generated in various situations. The **Response_Code** attribute, <RespCd>, is assigned from the values in Table 4.

If an [A3] cannot be signed due to a further failure⁸, then an unsigned [A4] is written in its stead. This has a message type such that it won't be read by the Counter, but it does capture the data for diagnostic purposes.

⁸ The [A4] generated in this case will contain the original Response_Code in <RespCd> and the new failure condition in <NewRespCd>.

⁹ Response_Code values of 30-39 and 90-99 have been allocated for "Failure by Agent". Other values are

Description	Blame	Response_ Code9	Comment
		30, 31	<i>Not relevant to NBX</i>
Failed by Agent: FI Timeout	FI	32	
		33	<i>Not relevant to NBX</i>
Failed by Agent: Invalid [A1]	FI	34	
		35	<i>Not relevant to NBX</i>
Failed by Agent: Signature Failure on [R1]	Horizon	36	
Failed by Agent: Invalid [R1]	Horizon	37	This includes failures in decrypting sensitive data and PIN Blobs
Failed by Agent: Operational Problem at Agent	Horizon	38	A catch-all
		39	<i>Reserved for future use</i>
Failed by Agent: FI Closed (Administrative)	FI	90 (LINK) 91 (A&L) 92 (CAPO)	Informed administratively of Closure or Half-Closure (see 5.1.3.3.3). The Agent_Error text will give information for the Clerk about the time (and date) of the expected resumption
		93, 94	<i>Reserved for future use</i>
Failed by Agent: FI Down (Operational)	FI	95 (LINK) 96 (A&L) 97 (CAPO)	Operationally detected that there is no connection to the PI
		98, 99	<i>Reserved for future use</i>

Table 4 – Response_Code Values for Agent-Detected Failures

5.1.3.1.3 Processing a Reversal

The Authorisation Agent will receive [C0] Counter-generated Reversals from two separate feeds:

- From the Routing Agent, that will have routed the [C0] to the Authorisation Agent that processed the [R1]. This route is unreliable because of its use of the Network from Branch to the Horizon Campuses.
- From the Guaranteed Reversals Agent via the C0 Reversals table in the NBX Persistent Store. This route is reliable but the [C0] may be delayed.

Before the [C0] from either feed is processed, its Digital Signature is checked (except where stated below). If it fails the check, the [C0] is logged but not processed. Unlike for [R1]s, this check cannot be suppressed.

For [C0]s received from the 'real-time' feed via the Routing Agent only, a staleness check will be performed. If more than 15 seconds old (configurable by the `StaleC0.Threshold` business parameter), the stale [C0] will not be forwarded to the FI and will not be logged. [C0]s from the 'guaranteed' feed are not subject to the staleness check.

The Authorisation Agent can also generate Reversal requests internally, in particular when it receives an [A1] Approve late.

reserved for other purposes.

The Authorisation Agent will use the Transaction Status information in the NPS to determine the processing appropriate to the Reversal request:

- If the Transaction cannot be matched against the Transaction Status table, it will not be processed. However, it will be logged to the Transaction Journal for feeding into the Transaction Enquiry System (TES).
- If the Transaction is sufficiently old that the Reversal should not be forwarded to the FI, the Reversal is logged but not forwarded. If it is so old that it is likely to have been purged from the Transaction Status table in NPS, the Reversal is discarded; otherwise, the Status information is updated to reflect that the Reversal had been requested. Two separate business parameters, `Reversal.Old.Threshold` and `Reversal.Old.MaxAge` respectively, control the two aspects of this behaviour though both are expected to be set to five days. *(The Threshold was introduced when it was thought that it would be set to 2 or 3 days rather than 5.)*
- If the Transaction has already been reversed or is in the process of being reversed, it will not be processed or logged. For efficiency, this check is normally made before the Digital Signature is checked.
- If the Transaction has been Declined, it will not be processed or logged. For efficiency, for a Reversal from the C0 Reversals table this check is made before the Digital Signature is checked.
- If the Transaction has been Approved, an [E1] Reversal is generated.
- If the [R3] has been sent but no [A1] received, the [E1] Reversal cannot yet be sent. It is held until the [A1] Approve is received, even if it is received after the timeout, at which point the [E1] is sent. The Transaction Status table is updated for the intended Reversal, which is also logged to the Transaction Journal for feeding into the TES for DRS; this is particularly relevant if the [A1] is never received.
- If no [R3] has been sent, as may happen if there was no available PI, the Reversal will not be processed or logged.

The Agent will use careful updates of the Transaction Status table to ensure that a Transaction cannot be reversed twice.

The [E1] Reversal is a 'must-deliver' message. If an [E2] Reversal Response is not received within a configurable period, an [E1] Reversal Repeat is sent. The `Reversal.Retry.Wait` and `Reversal.Retry.Max` business parameters for controlling this cycle.

The Transaction Status table in the NPS acts as the "store and forward" queue for these must-deliver messages. The Authorisation Agent normally relies upon an in-memory copy of this queue, but when an Authorisation Agent first becomes active it scans the Transaction Status table for outstanding Reversals.

If the elapsed time between sending the [R3] to the FI and making the first attempt to send the [E1] for a Counter-originated Reversal exceeds the `Reversal.Late` business parameter (typically 60 seconds), the Reversal is deemed to be 'late', and the Journal entry is marked accordingly.

The first [E2] Reversal Response received in a timely manner indicates that the FI has received the [E1] and no further attempts to send it will be made. Apart from logging the [E2] to the Transaction Journal, the [E2] is not processed. Subsequent [E2]s, and any [E2] received after a 60-second timeout (configurable by the `Timeout.FI.E2` business parameter), are ignored but will be logged for audit purposes.

Once a Reversal from the [C0] Reversals table has been secured in the Transaction Status table, the entry in the [C0] Reversals table is marked as 'actioned', i.e. it is logically deleted. It will be physically by an overnight NPS housekeeping process.

5.1.3.1.4 Message mappings

The message formats to be used with the FI_EEs are defined in the relevant Application Interface Specifications: [CAPO_AIS], [LINK_AIS] and [A&L_AIS]. These formats are all loosely based on different dialects of ISO 8583 (1987). The dialects for LINK and A&L are broadly very similar, with the most significant differences being in the use of Network management messages.

The mappings between internal Horizon formats and the external formats are defined in Mapping Documents: [CAPO_MAP], [LINK_MAP] and [A&L_MAP].

The Agents need not do any character set transformations when sending messages to and receiving messages from the FI_EEs. Everything is in ASCII. Many fields use a hexadecimal character representation of binary – in most cases the particular transformation is performed by Counter code.

The one exception to the use of ASCII is the representation of the Bitmap fields in CAPO messages, which are in binary rather than in hexadecimal characters. The use of the Tertiary Bitmap field is not supported if the AIS defines that Bitmaps are transferred in binary. (*None of the current AISs requires this combination of support.*)

5.1.3.1.5 Transaction Status

The **Transaction Status table** in NPS is used by the NBX Authorisation Agents to create and maintain the status of all the transactions. There is a separate instance of this table for each logical Authorisation Agent. The table is solely for use by the Authorisation Agent, though housekeeping is performed by the NPS.

The processing of a transaction is controlled by transition through a series of states. These are recorded in the transaction's CURRENT_STATUS in the Transaction Status table. Please refer to the Low Level Design [NBXAUTH] for details, where most of the transitions are shown in a series of state tables.

The columns YDDD + TERMINAL_ID + TRANS_NUM will form the primary key of the table. Almost all Agent access is directly through the primary key. All [R1], [A1], [C0] and [E2] messages contain the required key information.

The table will be partitioned on YDDD (Receipt Date). Transactions must be retained for at least five days, so at any one time there will be six partitions containing transactions. NPS housekeeping will remove older partitions and will also create new partitions for one or two days ahead.

For efficiency of access, each partition in the table will be sub-partitioned on the hash value of TERMINAL_ID. [NPS] states that there will be 64 sub-partitions (each containing approximately 22000 transaction for CAPO).

Whenever an NBX Authorisation Agent instance becomes the active Agent, its Reversals Management thread has to query the Transaction Status table for all the 'must deliver' transactions, that is to say all Reversals for which the CURRENT_STATUS indicates that a Reversal is still required. To facilitate this search, there will be a local index on the CURRENT_STATUS column.

5.1.3.1.6 Transaction Journal

The **Transaction Journal** in NPS is used by the NBX Authorisation Agents to audit (log) all messages passed across the external interface to the FI_EE, and to log various significant events. The spreadsheet embedded in [NBXJNL] identifies the different classes of journal record.

The Transaction Journal in the NPS provides a feed into the Transaction Enquiry Service (TES) of all the material events in the life of a transaction. Each transaction is identified by the Terminal_Id (the group and counter originating the transaction) and the Retrieval_Reference_Number (RRN). The RRN comprises a YDDD component from the

Receipt_Date, and the last 6 material digits from the Horizon_Txn_Num. Each 'transaction part' is identified by a sequence number that reflects the order in which they were inserted.

The Transaction Journal contains a separate column, in general, for each field that occurs in the transactional messages to the FI_EE. This is for the benefit of the TES extraction process (the TES-CO harvester) that populates the various tables in TES, which in turn is for the benefit of TESQA, the TES Query Application. The obvious exceptions are the fields that are sensitive, such as the card's Expiry Date. Please refer to the spreadsheet in [NBXJNL] for full details of these fields and for which message types these fields are relevant.

The Agents write various End of Day records into the Transaction Journal for reading by TES. When TES reads them, it will know that all the transaction records written before them have been flushed through to TES.

An End of Business Day (EOD) record is written at 8.20 pm each day. See the next section, 5.1.3.1.7, on Settlement for further information.

An End of Calendar Day (EOD_Cal) record is written soon after local midnight. It is written at the start of the new 'Journal batch' that is started at midnight (see Journal Management below), thereby ensuring that all transaction records written the previous calendar day have been flushed through. As a fail-safe, a record is also written on Agent start-up; this records the completion of the previous calendar day. This EOD record triggers a daily report used for statistical purposes only. *(Note: If no Agent runs for a complete midnight to midnight period, then there will be no record for the completion of the previous day.)*

In general, only the active Authorisation Agent inserts messages in to the Transaction Journal. The only messages inserted by the standby Agent are possible PI Unavailable messages during Agent start-up.

The Authorisation Agent never reads or updates a record once written. No other process writes to the table, though housekeeping will be performed by NPS.

The use of the Transaction Journal for systems management purposes is covered in 6.2.4.

5.1.3.1.6.1 *Journal types and subtypes*

Each journal record is categorised by its Journal_Type and Journal_Subtype. This two-level categorisation is primarily for the benefit of the TES extraction process, as the Journal_Type is mostly sufficient for it to determine how to process the record. Note that some Journal_Types do not require division into subtypes, and that a Journal_Subtype of "Rpt" (= Repeat) means that the record does not contribute to the business outcome of the transaction.

The list of Journal_Types is as follows. Other types may readily be introduced. Refer to the spreadsheet in [NBXJNL] for information on Journal_Subtypes.

Journal_Type	Category	Remarks
R1	[R1] message	The only [R1] messages that are audited in this category are those that would not otherwise be audited, i.e. those which are discarded because they are stale or a duplicate.
R3	[R3] message	Audited immediately after generating the [R3], before it is transmitted to the FI_EE
A3	Agent-initiated [A3] message	[R3]/[A1] timed out; or unable to generate the [R3]
A1	[A1] message	Audited soon after receipt of the [A1] from the FI_EE

C0	[C0] message	The only [C0] messages that are audited in this category are those where the [E1] cannot or should not be generated. Normally, the details of the [C0] are included in the log record for the [E1]
E1	[E1] message	Audited immediately after generating the [E1] Reversal, before it is transmitted to the FI_EE
E2	[E2] message	Audited soon after receipt of the [E2] Reversal Response from the FI_EE
EV	Events	Miscellaneous events
NMRQ_NBX	0800 from NBX	Network Management Requests from NBX to FI_EE
NMRQ_FI	0800 from FI_EE	Network Management Requests from FI_EE to NBX
NMRP_NBX	0810 from NBX	Network Management Responses from NBX to FI_EE
NMRP_FI	0810 from FI_EE	Network Management Responses from FI_EE to NBX
RJ_NBX	0620 from NBX	Reject (Administration Advice) from NBX to FI_EE
RJ_FI	0620 from FI_EE	Reject (Administration Advice) from FI_EE to NBX

Table 5: Journal_Types in the Transaction Journal

5.1.3.1.6.2 *Journal management*

For efficiency of harvesting the journal records into TES, journal records are written in 'batches'. Each record in a batch is assigned a three-level batch identifier comprising the current date batch, a physical partition number and a logical subpartition number within that.

The NPS is ultimately in control of the identifiers to be assigned to the 'next' batch, and provides the Agent with an interface to ensure that the next batch always has a batch identifier greater than any previously used.

Because of the multithreaded nature of the Agent, some threads will still be writing to the previous batch even after a new batch has been started. Therefore, NPS provides a separate procedure to allow the Agent to tell NPS that a batch (and all previous batches) has been completed and that it is now available for TES to harvest. A newly active Agent will, after a configurable delay, also commit any batches left over by a failed Agent. Together, these ensure that journal records are available to TES within 30-60 seconds of their being written.

The technique is described in much more detail in [NPS], as are the stored procedures that NPS provides to the Agents to manage the batches. The Agent configuration parameters controlling this are defined in 5.1.3.9.1.1.

5.1.3.1.7 Settlement

NBX is capable of acting as either Settlement master or Settlement Slave. However, configuration of the NBX Authorisation Agents via 'End of Day Cutover' business parameters (see [BUSPARAMS]) is independent of that of the Reconciliation File processing. The NBX Authorisation Agents will be configured to act as the Settlement Master with CAPO and A&L and as the Settlement Slave with LINK.

The NBX Business Day (or Settlement Day) runs from 8.00 pm the previous evening to 8.00 pm, configurable by the EODCutover.Time business parameter. The Settlement Date (bitref. 015) on all [R3] messages sent by NBX is that in force at the moment the [R3] was generated, which could be a moment or two before it was transmitted to the FI_EE. When NBX is the Settlement Master, it is this Settlement Date that is relevant to the reconciliation process.

The Settlement Date on the [A1] response message is only material when the FI_EE is the Settlement Master (i.e. for LINK). In this case it contains LINK's Settlement Date, which may be different from NBX's (and from the end bank's), and it is this Settlement Date that is relevant to the reconciliation process.

The Settlement Date on all [E1] Reversal messages sent by NBX is that on the [R3] being reversed.

An End of Business Day (EOD) event record is written to the Transaction Journal at 8.20 pm each day (configurable by the `EODCutover.EventTime` business parameter). This indicates that a sufficient period has elapsed since the change of Business Day at 8.00 pm. All [R3] messages for the Business Day just completed, and any Reversals pertaining to that or earlier Business Days, will have been flushed through to TES by the time that it reads the EOD record. For the benefit of TES the record contains both `EODCutover.Time` and `EODCutover.RevTime`, the latter being the cut-off time (8.10 pm) used for Reversals. *(Note that `EODCutover.RevTime` is not otherwise used by the Agents.)*

As the EOD record is critical to the REC file generation, its writing is recorded in the NPS Systems Parameters table. When an Agent first becomes active, it checks this system parameter to determine whether the previous day's event record has been logged. If not, it belatedly logs it.

When NBX is the Settlement Master (CAPO and A&L), this EOD record triggers the production of the reconciliation (REC) file. When NBX is the Settlement Slave (LINK), this triggers the processing of the reconciliation (LREC) file received.

Note that LINK also cut over their Settlement Date at 8.00 pm. The scheduling of LREC processing may go awry were this to change.

The cutover from one Business Day to another may be conveyed to the other party by means of a particular Network Management (0800) message, called the EOD Cutover message. Where defined by the AIS, it is sent from the Settlement Master to the Settlement Slave. Note, however, that the message is essentially irrelevant to the process of reconciliation. See 5.1.3.2.3.7 for further information.

5.1.3.2 Interfacing with the FI_EEs

Each NBX Authorisation Agent has a separate PI Handler for each of the FI_EE's PIs. Each PI Handler is realised by a single EE_IO Thread, backed by a corresponding FI Management Thread for processing Network Management messages and for journalising.

An [R3] message has to be allocated to a particular PI before it can be generated, as the encryption of the PIN Block depends upon the working key (AWK) specific to the selected PI.

The time (in ticks – so as to avoid any problems with clocks being changed) that the [R3] was written to the connection is retained for SLA and performance monitoring purposes, as is the time the corresponding [A1] is received.

Each [R1] message carries the Agent_Timeout value that should be used; the attribute is <AgtTmOut:> and is in milliseconds. This timeout applies to the time from which the [R3] message is sent to the FI_EE until the Agent stops waiting for an [A1] to be returned. The number of different Agent_Timeout values is likely to be very small, in practice only one. An indicative value is 18 seconds. Performance calculations are done on this basis. The code is not guaranteed to behave correctly if the value is greater than a configurable maximum, currently 30 seconds. *(Note that the value used to timeout the [A1] response is controlled by the Counter via an attribute in the [R1]. It is not controlled and not configurable within the Agent. The <AgtTmOut:> attribute is mandatory in the [R1].)*

The Agent also applies a timeout to [E2] Reversal responses, but this is only for internal housekeeping and has no material effect on the 'must deliver' properties of Reversals.

5.1.3.2.1 Load-balancing

The Agent is responsible for load balancing the [R3]s across the available remote Virtual Addresses (VAs). As stated earlier, each PI may form connections to one or more VAs – two in the case of CAPO. Furthermore, there could be multiple connections to any one VA, but that is not relevant to the load balancing decision. For development reasons, it has been decided to implement load balancing across TCP/IP connections rather than across VAs. In practice, as the current TISs have just one connection per VA, this meets the immediate requirement to load balance across VAs.

The [R3]s to be written are distributed amongst the connections on an essentially round-robin basis.

Each [R1] received from an NBX Routing Agent is assigned a per-Routing-Agent Message Sequence Number for the purpose of load balancing. This number, taken modulo the total number of connections across all the PIs, is used to select one from an ordered list of connections. The CAPO TIS calls for a specific ordering of connections, as per the following example. The PIs could be {PI_p, p = 0..3} with each PI_p having two connections {Q_{pq}, q = 0..1}. Then the order is {Q₀₀, Q₁₀, Q₂₀, Q₃₀, Q₀₁, Q₁₁, Q₂₁, Q₃₁}. Note that consecutive entries in this ordering are for different PIs.

If the selected connection is not currently Available and Responsive, the second choice uses a locally maintained sequence number, taken modulo the number of usable connections, to select one from the ordered list of usable connections. For fuller details of the algorithm, see the Low Level Design [NBXAUTH]. The selected connection (or rather an internal index for its socket) is remembered so that any Reversal can be sent to the same VA.

It may be that the Agent has only just started establishing connections and needs to be given time to get going. To prevent overloading the few connections that have been established, no second-choice allocations are made unless the number of connections has at least exceeded a threshold. This configurable threshold, `UseAltSocket.PercentMustBeOper`, is currently set to 50%, meaning that 4 sockets out of 8 for CAPO, and 1 out of 2 for LINK and A&L, must be operational for second-choice allocations to be used.

For Reversals, there may be restrictions as to which PI or VA to which the [E1] may be sent. Typically, the [E1] has to be sent to the same PI or VA to which the [R3] was sent. This is

controlled by the `Reversal.Rule` business parameter, which may be set to either `SameVA` (for CAPO) or `SamePI` (otherwise).

5.1.3.2.2 Managing the TCP/IP connections

The Technical Interface Specifications (TISs) define whether the Agent or the FI_EE's PI initiates the TCP/IP connections. NBX initiates the connections with CAPO and A&L; LINK initiates the connections to NBX.

Each NBX Authorisation Agent is capable of either behaviour. The actual behaviour is controlled by registry configuration. If a (set of) ports is configured for listening for inward connections, then the Agent will listen for inward connections and will not initiate outward connections. If such ports are not configured, the Agent will initiate outward connections – the (set of) virtual addresses to connect to are defined in registry, in `SOCKET_HOSTS` and `SOCKET_SERVICES`, which are resolved in the platform's Hosts and Services files.

The TISs also define the number of connections per PI and the use of Virtual Addresses (VA = IP address plus port number). This information is also configured via registry. See 5.1.3.9.2 for further information on configuration by registry.

Only the active Agent of a Resilient Agent Pair forms connections with the FI_EE. The standby Agent does not initiate outward connections, does not listen for inward connections from the FI_EE, and does not listen for health probes.

5.1.3.2.2.1 *Connecting outwards*

Outward connections are configured only for A&L and CAPO. For A&L the Agent is configured for one TCP/IP connection per PI. For CAPO the Agent is configured for one connection to each of two VAs, with each VA being serviced by a separate 'remote thread'.

The Agent is capable of being configured (per PI) with multiple connections to a single VA or with single connections to multiple VAs. *(The NBX Authorisation Agent has a limitation that it does not cater for multiple connections to each of multiple VAs.)*

Configuration of Host and Service names

The PI Handler Threads are configured (collectively via registry) with the set of host and service names for the Virtual Addresses of the FI_EE's PIs. The registry values for these names contain substitution parameters, substituted at run-time (not build-time):

- %I for the Agent's Identity (e.g. CAPO_A)
- %T for the name of the target PI from a list of such names
- %V for an index (1, 2, ...) of the Virtual Addresses within the target PI

The names, thus expanded, are resolved in the Hosts file into an IP address and in the Services file into a port number. The registry values are:

- `SOCKET_HOST` = %I_%T
- `SOCKET_SERVICE` = %I_%T_VA%V (for CAPO) or %I_%T (for A&L)

Thus the Hosts file would need to contain, for example, CAPO_A_PIA1, CAPO_A_PIB1, CAPO_A_PIC1 and CAPO_A_PID1 for the CAPO_A Agent to connect to the four PIs: PIA1, PIB1, PIC1, and PID1. Similarly, the Services file would need to contain CAPO_A_PIA1_VA1 and CAPO_A_PIA1_VA2 for the CAPO_A Agent to connect to PIA1.

5.1.3.2.2.2 *Inward connections*

Inward connections are configured only for LINK.

The Agent listens on the configured set of ports, one port per PI. When an inward connection is received on the port, it is accepted. If the number of accepted connections now exceeds the configured maximum of one, the previous connection is aborted (until Release S90). From S90, the Agent behaviour will be changed and the new connection will be rejected. However, it will not be rejected immediately. Instead, the Agent will hang on to the new connection for

up to 40 seconds (configurable by registry), long enough to give every chance of an original, stale connection disappearing. The delay will also help thwart a denial of service attack. [PC0111214]

(The NBX Authorisation Agent has a limitation that restricts the socket concurrency to one per PI for inward connections.)

Configuration of Service names for listening

The PI Handler Threads are configured (collectively via registry) with the set of service names for the ports for inward connection. The registry values for these names contain substitution parameters, substituted at run-time (not build-time):

- %I for the Agent's Identity (e.g. LINK_A)
- %T for the name of the PI from a list of such names

The names, thus expanded, are resolved in the Services file into a port number. The proposed registry values are:

- LISTENS = <EE_IO:<Port:TMSNX_%I_%T>>

Thus the Services file would need to contain TMSNX_LINK_A_PIA1 and TMSNX_LINK_A_PIB1 for the LINK_A Agent to listen for PIA1 and PIB1.

5.1.3.2.2.3 Health probe for inward connections

NBX offers LINK a single virtual address per PI for LINK to connect to, no matter at which campus the active Agent is running. The NBX service is virtualised using a Content Switching service (see [LINK_TIS] and [NBXNETWORK]).

The Content Switch probes the Agents at both campuses to determine which one is the active instance and which the standby, and routes inward connections from LINK to the active instance. The active Agent listens on a special port configured for the health probe, the standby Agent does not. It is this difference in behaviour that allows the Content Switch to determine which is which. See section 5.4.2 for more details.

The health probe continues to operate even when the FI_EE is administratively Closed, in order that the Content Switch will not raise any alerts in the absence of an active Agent responding.

Configuration of Service names for health probe

The PI Handler Threads are configured (collectively via registry) with the set of service names for the ports for health probes. The registry values for these names contain substitution parameters, substituted at run-time (not build-time):

- %I for the Agent's Identity (e.g. LINK_A)

The names, thus expanded, are resolved in the Services file into a port number. The proposed registry values are:

- LISTENS = <EE_Probe:<Port:TMSNX_%I_CHECK>>

Thus the Services file would need to contain TMSNX_LINK_A_CHECK for the LINK_A Agent to listen for the health probe.

Note that the configuration of this health-probe port is completely independent of that of the main listening ports used for inward connections.

5.1.3.2.2.4 Disconnecting

Either party may initiate the disconnection, no matter which party initiated it.

When the Agent disconnects, it distinguishes between a graceful disconnection and an abort. In circumstances specified below, the Agent behaviour is controlled by the `Disconnect.Graceful` business parameter.

With LINK the distinction is important. If a connection is disconnected gracefully LINK will not automatically attempt to reconnect; operator intervention will be required, so this option should be used only prior to a period planned of maintenance. On the other hand, if a connection is aborted LINK will automatically attempt to reconnect for up to a period as specified in the TIS.

The Agent performs a graceful disconnection:

- following a Logoff (not command-initiated) and `Disconnect.Graceful` is 'Y'
- when ceasing to be the active Agent and `Disconnect.Graceful` is 'Y'
- in response to an `Admin_Close` operator command (see 5.1.3.3)

The Agent aborts a connection in all other circumstances, which include:

- following a Logoff (not command-initiated) and `Disconnect.Graceful` is 'N'
- when ceasing to be the active Agent and `Disconnect.Graceful` is 'N'
- following any failure on the socket servicing the connection
- following acceptance of a new inward connection – the previous connection, if any, for this PI is aborted (see 5.1.3.2.2.2)
- in response to a `Reset_Connections` operator command – tantamount to the operator informing the Agent that the socket has failed (see 5.1.3.3)

A graceful disconnection is implemented by a **shutdown** command to the socket followed by **closesocket**. An abort is implemented by a **closesocket** command alone. The other party may also have initiated the graceful disconnection, in which case the Agent simply uses **closesocket** to free local resources. Socket options are set such that there is 'no linger' on the shutdown.

5.1.3.2.2.5 *Reporting status of the TCP/IP connections*

The availability of individual TCP/IP connections, excluding any related to the health probe, is reported identically in both the Transaction and Management Journals, using a `Journal_Type` of "TCP_STS". The status of the connection is recorded as "Avail" or "Unavail". The connection is identified by the PI it relates to and a 'socket index' within that PI.

The sockets used for listening in the PI Handler (EE_IO) and health probe threads are opened at the time the Authorisation service itself becomes available. This service availability is recorded in a MONID event (see Table 29); there is no separate Journal message indicating the start of listening.

5.1.3.2.3 Network Management messages

The NBX Authorisation Agents are capable of supporting the Network Management (0800) messages in the following table (Table 6). All these messages operate at the PI level, except for Handshakes that are used on individual TCP/IP connections.

The messages actually supported by a particular NBX Authorisation Agent, and their associated Network Management Information Codes (NMICs), are configured by business parameters. For example, the most restrictive support is for CAPO, where CAPO initiates none of these messages and the Agent initiates only Log On, Log Off, Key Change and Handshake messages.

Each message exchange consists of an 0800 message initiating the action and an 0810 response message confirming or denying the action.

The STAN (bitmap ref. 011) on 0800 (and 0620) messages is generated independently of those used on [R3] and [E1] messages. The first STAN for a PI after the Agent goes active is derived from the clock. Subsequent STANs go up in twos, in such a way that the Agents on primary and secondary servers use disjoint sets.

Network Management message	Capability as initiator	Capability as responder	Purpose
Log On (<i>or</i> Sign On)	Yes	Yes	Establishes a 'PI session'
Key Change	Yes	Yes	Initiates a change of AWK
Key Change Request	Yes	No	Requests the other party to initiate a change of AWK
Online Key Verification	No	Yes	Requests the other party to an AWK
Handshake (<i>or</i> Echo Test)	Yes	Yes	Tests if other party is able to respond
Log Off (<i>or</i> Sign Off)	Yes	Yes	Terminates a PI session
EOD Cutover	Yes	Yes	Notifies change of business day (and hence of settlement date)

Table 6 – Network Management (0800) messages: Agent capability

The description below is written in terms of the following state variables:

- Logged On / Logged Off: indicates whether a PI session has been established
- Available / Unavailable: indicates whether a PI session has been fully established, i.e. that an AWK has been agreed
- Responsive / Unresponsive: indicates whether an Available session is free of problems

When selecting a possible TCP/IP connection for sending an [R3], only Available and Responsive connections are considered.

5.1.3.2.3.1 Log On

A Log On message is used to establish a PI session. It may also be used to 'reset' a session in case the two parties have a different view as to whether there is a current PI session, as could happen when the two parties have a different perception on the order or nature of incidents on the TCP/IP connection(s).

The Agent initiates a Log On:

- when the first or only connection is established
- when the lack of Handshake responses has triggered a Log On to 'reset' a session (see 5.1.3.2.3.5)
- in response to a Logon operator command (see 5.1.3.3)

LINK and A&L may also initiate a Log On. This may be at any time, but LINK is expected to initiate a Log On when the first (and only) connection is established. A&L may do this.

To avoid unnecessary collision of Log On messages when either party can initiate the process, the Agent may be configured to delay initiating it (*SignOn.Delay* business parameter, set to 3 seconds for LINK in accordance with the TIS).

The Agent enters a Logged On state for this PI on the first of:

- receiving a Log On response (effectively ignored if already Logged On)
- sending a Log On response (sent even if already Logged On)

If no response is received (or if the response is a Nack) and the Agent is still not in a Logged On state, another attempt to Log On is made. This is controlled by the *SignOn.Wait* and *SignOn.Max* business parameters. If the retry limit is reached, all TCP/IP connections are disconnected. Note that the Logon operator command is a single shot command and retries are not attempted, nor is the connection disconnected in the event of failure.

A successful Log On establishes a PI session. However, the session is not 'fully established' (and the PI deemed to be Available) until an Acquirer Working Key (AWK) has been agreed. This means that the PI is necessarily Unavailable until then.

5.1.3.2.3.2 *Key Change*

Only one party, as defined by the AIS, may initiate a Key Change. The Agent's behaviour is controlled by the `KeyChange.NBXMaster` business parameter.

The NBX Crypto subsystem imposes a minimum interval between key changes. When NBX is the master, the Agent ensures that it does not initiate key changes more frequently than its own minimum key change interval, `KeyChange.MinInterval` (typically 2 minutes), which is set at least as large as crypto's minimum.

NBX is master

(Only for CAPO.) If NBX is the key change master, the Agent initiates a Key Change:

- when first Logged On to establish an AWK
- upon receipt of n consecutive invalid PIN block responses (in [A1] messages) in a session
- every n days (e.g. every day) within a session
- in response to a `Change_AWK` operator command (see 5.1.3.3)

The `AWKError.Limit` (typically 6) and `AWKError.Response` (76) business parameters control the change of AWK for consecutive PIN block failures. A PIN block failure does not contribute to this count if it occurs less than the minimum interval between key changes (`KeyChange.MinInterval`) since the last key change. This enables the Agent to ignore all further PIN block failures being reported against the old AWK.

The `KeyChange.Time` and `KeyChange.MinAge` business parameters control the periodic key change. `KeyChange.Time` is the local time (typically 2.30 am) for the daily check for considering a key change. If, at key check time, the key has been changed more recently than `KeyChange.MinAge` (typically 80 minutes), no key change will be initiated.

The payload of the Key Change message is the Network Management Information field (bitmap ref. 125). This is generated by calling the crypto function `NbxGenerateAwk` (see 5.5.3.1).

If the response is a key change failure (i.e. `KeyChange.Nack.RespCd`, 76), another attempt is made using the same new AWK. This is controlled by the `KeyChange.Retry.Interval` and `KeyChange.RespFail.Max` business parameters. If this retry limit is reached, the retry count is reset and a further sequence of attempts is made with another AWK. The number of such sequences is limited by the `KeyChange.RespFail.AWKMax` business parameter. If this further retry limit is reached, an alert is raised (as an Error in the NT Event Log) and the key remains unchanged.

If no response is received (or if the response is a Nack other than as above) another attempt is made using the same new AWK. This is controlled by the `KeyChange.Retry.Interval` and `KeyChange.NoResp.Max` business parameters. If the retry limit is reached, an alert is raised and the key remains unchanged.

Note that the `Change_AWK` operator command is a single shot command and retries are not attempted.

The payload (i.e. the Response Code) on the Key Change response is confirmed with the crypto subsystem by calling `NbxConfirmAwk` (see 5.5.3.1).

NBX is slave

(Only for LINK and A&L.) If NBX is the key change slave, it is the FI_EE's PI that initiates a Key Change. When first Logged On, it is the expected behaviour, according to the AISs, that LINK and A&L will immediately initiate a Key Change (see 5.1.3.2.3.3 below if they fail to do so). The PI also initiates a Key Change in similar circumstances to NBX. Note that when NBX is slave it is the PI, not NBX, that is responsible for counting consecutive invalid PIN block responses and initiating the Key Change (though NBX does count them for statistical purposes).

When the Key Change message is received, the payload in the Network Management Information field (bitmap ref. 125) is passed to the crypto function **NbxReceiveAwk** (see 5.5.3.1) for verification. If the verification fails, the Key Change response will contain a suitable Response Code – this is normally returned by the crypto function but exceptionally it is completed from the `KeyChange.Nack.RespCd` business parameter (set to 76).

New AWK is agreed

A successful agreement on an AWK means that the PI session is now fully established and the PI deemed to be Available. If the PI was previously Available, i.e. there was already an agreed AWK, this AWK continues to be used until superseded.

5.1.3.2.3.3 Key Change Request

A Key Change Request message requests the other party to initiate a Key Change.

NBX is slave

If NBX is the key change slave, the Agent initiates a Key Change Request:

- when Logged On for a while and no AWK has been agreed (controlled by the `KeyChange.Request.Wait` business parameter, perhaps 10 seconds) – this is a long-stop and should not happen
- every *n* days (e.g. every day) within a session
- in response to a `Change_AWK` operator command (see 5.1.3.3)

The Key Change Request is not specifically retried, but note that if the first condition persists then retries will, in effect, occur. There are no business parameters for retries.

The periodic key change is controlled as above for Key Change, except that there is another business parameter, `KeyChange.SkipAge` (typically 22.5 hours). If, at key check time, the key was last changed more than this 'skip age' ago, no key change will be initiated. This is on the basis that the other party will initiate the key change after 24 hours, and we wish to avoid both parties initiating a new AWK simultaneously.

5.1.3.2.3.4 Online Key Verification

An Online Key Verification message requests the other party to test the AZMK by processing a test AWK. Only LINK will initiate one, its main purpose being to verify that a new Acquire Zone Master Key (AZMK) has been successfully promoted by both parties.

When the Online Key Verification is received, the payload in the Network Management Information field (bitmap ref. 125) is passed to the crypto function **NbxKeyTestCheck** for verification. If the verification fails, the response will contain a suitable Response Code – this is normally returned by the crypto function but exceptionally it is completed from the `KeyChange.Nack.RespCd` business parameter (set to 76).

5.1.3.2.3.5 Handshake

A Handshake message invites the other party to respond, thereby enabling the detection of the other party's inability to respond, i.e. to detect faults that do not involve connection failure. No inferences can be made from not receiving Handshake messages, only from not receiving responses to Handshake messages. CAPO do not initiate Handshakes.

NBX is initiator

The Agent initiates a Handshake:

- when nothing (apart from Handshake messages) has been received on a TCP/IP connection for a while
- when a response has been timed out, as this may indicate a problem
- in response to a Handshake operator command (see 5.1.3.3)

The first of these is controlled by the `Handshake.Idle.Interval` business parameter (typically 60 seconds).

If no response is received, retries are controlled by the `Handshake.Retry.Wait` and `Handshake.Retry.Max` business parameters (typically 15 seconds and 2 to 5 retries). If the retry limit is reached, the action taken is controlled by `Handshake.EventFlag` and `Handshake.SignOnFlag`. These are set in accordance with the relevant TIS.

The CAPO and A&L TISs require that NBX automatically takes recovery action. `Handshake.SignOnFlag` will indicate that the recovery action is to initiate a Log On sequence (see 5.1.3.2.3.1) – note that the temporary Unavailability while this is happening affects all the TCP/IP connections for this PI, not only the one on which Handshakes failed. If the Log On sequence fails to re-establish a PI session, then its failure action is to disconnect all the TCP/IP connections, again not only the one on which Handshakes failed.

The LINK TIS requires that NBX takes no automatic recovery action but simply raises an alert. The expectation is that the fault will lie between LINK's TCP/IP handler and the PI itself, so that the above recovery actions will have no effect. The alert should cause the NBX operator to contact his LINK counterpart to investigate. Note, however, that if the fault is as expected, then the LINK operator will already be aware of the problem. Having raised the alert, the Agent will start the Handshake sequence again. It will also mark the affected connection as being Unresponsive, and the Agent's load balancing will avoid scheduling [R3]s to that connection.

Note that the Handshake operator command is a single shot command and retries are not attempted, nor is any consequential action taken in the event of failure.

NBX is responder

NBX simply responds to the Handshake message. The Response Code always indicates success. The response is sent even if not Logged On (and even if unexpectedly received from CAPO!).

5.1.3.2.3.6 Log Off

A Log Off message terminates a PI session. There is no relationship between which party initiates the Log On and which the Log Off. With CAPO, only NBX can log off.

NBX is initiator

Most logoffs are implicit: e.g. when an active Agent collapses, or when a Log On is used to reset a session. Thus the Agent only initiates a Log Off in exceptional circumstances:

- in response to a Logoff or Admin_Close operator command (see 5.1.3.3)

The Agent waits a short while for the Log Off response, controlled by the `SignOff.Wait` business parameter (typically 5 seconds). If it is not received the Log Off is not retried, and the logging off procedure continues normally.

The state becomes not Logged On and the PI session Unavailable immediately the Log Off message is initiated.

NBX is responder

LINK and A&L may also initiate a Log Off. The Agent always responds with a response indicating success, even if the Agent considers itself not to be Logged On. The state becomes not Logged On and the PI session Unavailable.

TCP/IP connections

There are different scenarios:

- When NBX is the responder, the Agent does not terminate the TCP/IP connections and will not automatically follow this with an attempt to Log On.
- With a Logoff command, the Agent similarly does not terminate the TCP/IP connections and will not automatically follow this with an attempt to Log On.
- With an Admin_Close command, the TCP/IP connections are gracefully disconnected.
- When no operator command is involved, the TCP/IP connections are either aborted or gracefully disconnected depending on the Disconnect.Graceful business parameter.

5.1.3.2.3.7 EOD Cutover

The EOD Cutover message is used to convey to the other party the fact that the initiator has cut over from one business day to the next and that therefore the Settlement Date has moved on to the new business day. It can only be sent by whichever party is the Settlement Master, i.e. the party that generates the daily REConciliation file.

The message is essentially irrelevant to the process of reconciliation:

- With LINK, who are the Settlement Master, LINK have stated, “whether or not we receive a response back, LINK will cut over to the next Settlement Day and create LREC files accordingly. We will not send another Cutover message on a PI for which no response was received.”
- With CAPO, where NBX is the Settlement Master, no EOD Cutover message has been defined. NBX will cut over to the next Settlement Day and create the REC file accordingly.
- With A&L, where NBX is also the Settlement Master, A&L require that the Cutover message is treated as a ‘must deliver’ message. Whether or not NBX is able to send the message and whether or not it receives a response, NBX will cut over to the next Settlement Day and create the REC file accordingly.

The EODCutover.NBXMaster and EODCutover.MsgFlag business parameters control whether NBX is the Settlement Master or not and whether an EOD Cutover message is involved.

The EODCutover.Time business parameter (typically 20:00:00) controls when NBX cuts over to the new NBX Business Day – see 5.1.3.1.7.

NBX is initiator

(Only for A&L.) The Agent initiates a EOD Cutover:

- at the change of Business Day
- when the PI first becomes Available and a check of the system parameter (see below) shows that the previous day’s EOD Cutover message has not yet been sent successfully
- in response to a EOD_Cutover operator command (see 5.1.3.3)

When a successful response is received it records the fact in the NPS Systems Parameters table.

If no response is received (or if the response is a Nack), another attempt to send the EOD Cutover is made. This is controlled by the EODCutover.Wait and EODCutover.Retry.Max business parameters. If the retry limit is reached, an alert is raised. Note that the EOD_Cutover operator command is a single shot command and retries are not attempted.

NBX is responder

(Only for LINK.) NBX simply responds to the EOD Cutover message. The Response Code always indicates success.

5.1.3.2.4 Reporting Availability of the PIs

The availability of the PIs is reported through four distinct channels:

- in Heartbeats (column EE_BAD: Y/N for each PI, but normally condensed to a single Y/N), for OMDB (see 6.2.1.2)
- in MONID event messages, for OMDB (see 6.2.3)
- in the Transaction Journal, for TES (see 6.2.4)
- in the Management Journal, for OMDB (see 6.2.4)

The reporting to the two Journals is identical.

The reporting of Unavailability through Heartbeats and through MONID event messages is delayed so that transient glitches do not cause unnecessary alerts. The delay is configurable by registry (typically 10 seconds).

The concept of **Availability** of a PI is the same for each channel: that there is a fully established session with the PI using an agreed AWK. Heartbeats continually record the current availability, whereas the other channels record changes in the availability. However, with the PI Status records (Journal_Type = "NMPI_STS") in the Journals, Availability has three states rather than just two:

- Journal_Subtype = "Avail"
- Journal_Subtype = "Unavail_PI" : reason does not attach to NBX
- Journal_Subtype = "Unavail_NBX" : reason attaches to NBX

The PI is reported as Avail:

- a) when an AWK has just been agreed

The PI is reported as Unavail_PI:

- b) when a Log Off message is received from the other party
- c) when a logon is initiated because of no response to consecutive Handshakes (i.e. only if `Handshake.SignOnFlag` is set)
- d) when the only or last TCP/IP connection has been lost (i.e. disconnected by the other party or by the network), or when there is a failure of such a connection

The PI is reported as Unavail_NBX:

- e) when a Log Off is initiated by the Agent (in practice, only by operator command)
- f) when the connections for a PI are reset by operator command
- g) when the only or last TCP/IP connection is being disconnected following Agent closure (Agent resigning, net stop, etc). This should be journalised before the Heartbeat that announces that the Agent is no longer active
- h) during Agent start-up, if the most recent 'active' Heartbeat written by either Partner is 'old' – this is reported for all PIs. This check has to be made before the Agent overwrites any previous Heartbeat with its first new Heartbeat
- i) on becoming the active Agent, if the most recent 'active' Heartbeat written by its Partner is more recent than any journalising resulting from the previous case – this is also reported for all PIs
- j) on any other transition from the Available state (if there are any)

The attribution of blame to NBX or to some other party is necessarily rough and ready, but it does give support staff a first indication prior to investigation. The categorisation treats network faults as non-NBX.

These records in the Transaction Journal are used in the PI Availability Report (see [TESPOREPORTS]) to determine periods of unavailability, measured from when the PI is recorded as unavailable to when it is recorded as available again. When an active Agent fails in such a way that it cannot itself journalise the unavailability, it is necessary for another Agent instance to record when the PI is deemed to have become unavailable. This time is taken from the last Heartbeat written by the failed active Agent.

As it is necessarily impossible to ensure that the start of a period of unavailability is journalised precisely once, the design errs towards journalising it more than once rather than not at all. (Note that it is possible for unavailability not to be recorded at all, but only when the period of unavailability is much less than the critical '1 minute' that triggers investigations of the cause. This can happen when EACRR restarts a failed active Agent so quickly that its most recent Heartbeat is still fresh.) The PI Availability Report will measure the period of unavailability from the first of a consecutive group of such records (where 'first' refers to the order the records are journalised).

Case h) covers the case where no standby Agent took over in a timely manner from a failed active Agent. Case i) covers the case where the standby Agent unilaterally took over from a failed active Agent. Case g) covers the case when an active Agent stood down.

As a fail-safe, in cases h) and i) unavailability is recorded for all PIs and not only those shown as available in the relevant Heartbeat.

5.1.3.3 Operator commands

The various parts of requirement NBR0535 call for an NBX "Console" Interface. (*I'm interpreting NBR0535b to require the retransmission only of Network Management request messages but not of response messages – the remote PI can send the request again if need be.*)

The supported commands fall into three groups:

- actions on a PI
- actions concerning Reversals
- administrative Closure of an FI_EE

The NBX Authorisation Agents will poll the **NBX Operator Commands** table in the NPS, with one command per row. This table will be common to all the Agents. All commands will be directed to the NBX Authorisation Agent for a specific Logical FI, e.g. CAPO_B, with each Agent selecting only those commands directed to itself.

It is envisaged that some sort of Tivoli task will capture the commands from an NBX operator and insert them into the table.

Each Agent will process the commands one by one in chronological order of insertion. Once a command has been read, and where necessary secured, a Processed flag will be set in the row entry in the table. An NPS housekeeping function can remove Processed rows overnight (c.f. similar behaviour with the C0 Reversals table.)

The commands that have to be secured before marking the entry as Processed are:

- those concerning Reversals, where the changes are secured by updating affected transactions in the Transaction Status table
- those concerning administrative Closure, where the changes will be secured in the NPS System Parameters table

Only the active Agent of a resilient pair will process commands.

As stated above, all commands are directed to the NBX Authorisation Agent for a specific logical FI. The following table does not explicitly mention this parameter, common to all commands.

The command names in the first column are case-blind. The Agent does not validate the command names. Instead, the thread that polls the Commands table will advertise the command and a thread that is responsible for actioning a specific command will notice relevant advertisements. If no thread has processed the command within a timeout period (configurable in registry), the advertisement is withdrawn and the entry in the Commands table is marked as Processed with an appropriate diagnostic.

Command	Parameters	Description
Logon (to a PI)	PI name (e.g. PIc2)	Requests immediate sending of a Log On message (0800/071) to the named PI. This will be sent even if the NBX Agent believes it is already logged on and will always result in a new AWK. The command will have no effect when the FI_EE is administratively Closed – it will not lift this Closure. If there is no TCP/IP connection the NBX Agent will already be trying to establish one, or, for LINK, will be advertising its availability for LINK to establish one. Once one is established the Agent will automatically attempt to log on, so there is no requirement for the Agent to remember this command if it is not immediately actionable.
Logoff (from a PI)	PI name	Requests immediate sending of a Log Off message (0800/072) to the named PI if there is a TCP/IP connection to send it on. It will remain logged off until one of: <ul style="list-style-type: none"> ▪ it is overridden by a Logon command ▪ the remote PI initiates the log on (cannot happen with CAPO) ▪ an administrative Closure is lifted The command has no effect on the TCP/IP connections themselves – they will not be disconnected gracefully nor aborted. A separate command will need to be given to effect this.
Reset_Connections (with a PI)	PI name	Requests that any and all TCP/IP connections to the named PI be aborted. This command does <u>not</u> result in any attempt to log off first. The NBX Agent will then try to establish a new connection (or connections), or, for LINK, will advertise its availability for LINK to establish one. Whether the new connection requires a new Logon is controlled by the same rules as when the connections are aborted in other circumstances.
Change_AWK (for a PI)	PI name	Requests immediate sending of a Key Change message (0800/161) where NBX is the key change master (i.e. CAPO) or of a Key Change Request message (0800/181) where NBX is the key change slave (i.e. LINK, A&L).
EOD_Cutover (of a PI)	PI name	Requests immediate sending of an EOD Cutover message (0800/261). For CAPO and LINK, there is no such message and the command has no effect.

		As an EOD Cutover message has no Settlement Date in it, the effect on the remote PI of sending another EOD Cutover message is not apparent.
Handshake (on a PI)	PI name	Requests immediate sending of a Handshake message (0800/361 for CAPO, 0800/371 for LINK, A&L).
Retry_Reversals (not PI-specific)	From date/time (optional) To date/time (mandatory and in the past)	Requests that all Reversals that arrived in NBX within the specified date range for which NBX had exhausted the retry algorithm are resent (as 0421 messages). Any timer and retry count associated with repeat attempts to send an [E1] are reset. The NBX Authorisation Agent finds these Reversals by scanning the Transaction Status table for transactions in states Reversal_Expired and whose Routing_Agent_Tsmp is in the specified range.
Stop_Reversals (not PI-specific)	From date/time (optional) To date/time (mandatory and in the past)	Requests that no more attempts are made to send any Reversals that arrived in NBX within the specified date range and that have not yet been successfully sent to the FI_EE. The NBX Authorisation Agent finds these Reversals by scanning its cache of Reversals that have not yet been sent.
Admin_Close (not PI-specific)	From date/time (optional) To date/time (optional and in the future) Time text (mandatory)	Requests that the Administrative Status of the FI_EE is set to Closed. If the 'From date/time' is omitted, this is effective immediately. If the 'To date/time' is omitted, this is effective indefinitely. The 'Time text' will be included in the dialogue message to the Clerk. Any working sessions with the FI_EE will be logged off, and the connections gracefully disconnected. No new connections will be re-established and, for LINK, though the NBX Agent will continue to advertise its availability through the health probe.
Admin_Open (not PI-specific)	None	Requests that the Administrative Status of the FI_EE is set to Open, i.e. that any actual or pending administrative Closure or Half-Closure be lifted immediately. The NBX Agent will attempt to establish new connections. For LINK, the LINK operator will need to initiate them.
Admin_Half_Close (not PI-specific)	From date/time (optional) To date/time (optional and in the future) Time text (mandatory)	Requests that the Administrative Status of the FI_EE is set to Half-Closed, i.e. that no [R3] messages are to be sent to the FI_EE. If the 'From date/time' is omitted, this is effective immediately. If the 'To date/time' is omitted, this is effective indefinitely. If the 'To date/time' is present, the Administrative Status is automatically changed to indefinitely Closed at the termination of the Half-Closure. The 'Time text' will be included in the dialogue message to the Clerk (who will not be able to distinguish this case from administrative Closure). During the Half-Closure all other interactions with the FI_EE are unaffected, including the sending of [E1] Reversals.

Table 7 – Operator commands to the NBX Authorisation Agents**5.1.3.3.1 Actions on a PI**

Most of the commands in this group allow the operator to instigate the sending of a Network Management (0800) message. Each of these act as a single shot command, in that no retries are attempted in the case of no response or a failure response. There is one exception: the CAPO AIS specifically requires that the Change_AWK command goes through the full retry sequence.

The remaining command allows the connections to be reset.

None of the commands in this group should be used without express co-operation and agreement between the operators of NBX and of the FI. They are only for use in extremis. It is possible for them to have a damaging effect: for example, if a Logon command for a PI fails the PI will become Unavailable even it was Available beforehand.

The effect of the command can be observed through OMDB's view on the Management Journal. This records all Network Management messages sent and received, the (Un)Availability of the PI and the state of the TCP/IP connections. The effect will also be observable by the FI's operators and this is expected in practice to be the normal method of observation.

5.1.3.3.2 Actions on Reversals

Reversals are must-deliver messages. The Agent retries sending a Reversal up to a configurable limit of attempts. Attempts to send the Reversal are only counted when there is a fully established connection on which to send it. When the limit is reached, the Reversal is still retained (up to 5 days), in the Transaction Status table, with a specific Status (Reversal_Expired) indicating this.

The two operator commands allow such expired Reversals to be tried yet more times, and conversely to 'expire' Reversals which would normally be subject to further retry attempts.

5.1.3.3.3 Administrative Closure

The migration strategy calls for the concept of administrative 'Closure' on a per FI basis. It also calls for the ability to suspend sending [R3] messages, a state termed 'Half-Closure' – this is particularly required when regressing back from NBX to NBE operation, where it is desirable to be able to allow a wash-up period during which Reversals may continue to be sent to the FI_EE.

Each of the three commands concerning administrative Closure completely supersedes any previous command in this group. The latest command for an FI_EE is remembered in the NPS System parameters table, so that it remains accessible to all Agent instances until it has effectively expired.

When an [R3] is received whilst the FI_EE is Closed or Half-Closed, an [A3] is generated with a Response_Code of FI_EE Closed (the actual value depends upon the FI, see Table 4). The associated Agent_Error text will be the time (and date) of the expected resumption, exactly as input as 'time text' to the Admin_Close or Admin_Half_Close operator command (e.g. "08:00 on Friday"). This text will be included in the dialogue message displayed to the Clerk. Note that 'time text' is limited to 32 characters.

5.1.3.4 Security**5.1.3.4.1 Spoofing**

Genuine [R1]s can only have been written at a Counter. To prevent spoofing using priority messages written at the Correspondence Server, the Agent checks the message's Node Id. Any message written at a Correspondence Server node will be ignored, and the event logged

as an Error. *(Note that messages imported at the Correspondence Server using the Riposte import facility cannot be priority messages, so will not appear on the priority real-time message port.)*

This check is performed by the Authorisation Agent rather than by the Routing Agent to detect a process masquerading as a Routing Agent.

The check can be overridden on test systems – see 7.1.1.1.

5.1.3.4.2 Verifying the Digital_Signature

Both [R1] and [C0] messages are digitally signed. Their signatures have not been checked by either the NBX Routing or Guaranteed Reversals Agents.

The Digital Signature attribute, <DSig>, signs the <RACMess> compound attribute in its Riposte Attribute Grammar (RAG) form. The processing steps are as follows:

- The value of the <DSig> attribute is in Base64-encoding. This is converted back to binary.
- The RAG string and the binary signature are passed to the crypto subsystem for verification (see 5.5.3.2.1). The verification is software-based and is mill-intensive.

Validation failures, including a missing Signature, cause a Failure-[A3] message to be returned to the Counter.

5.1.3.4.3 Decrypt the Encrypted Data

The Encrypted Data, <Encrypt>, is an attribute within the Security Data, <Sec>. If it is present and not empty (it should always be present and not empty – however, no error should be generated if this is not the case), the processing steps are as follows:

- The value from the [R1] <Encrypt> attribute is in Base64-encoding. This is converted back to binary.
- The binary data is passed to the crypto subsystem for decryption (see 5.5.3.2.3). The decryption is software-based and is mill-intensive.

Failures cause a Failure-[A3] message to be returned to the Counter.

5.1.3.4.4 Translate the PIN Blocks

The PIN Blobs are attributes, <PIN_n>, within the Security Data, <Sec>. For each PIN Blob present (from zero to two, but normally no more than one), the processing steps are as follows:

- The value from the [R1] < PIN_n > attribute is converted from Base64-encoding to binary.
- The binary data is passed to the crypto subsystem for verification, decryption and re-encryption ready for the [R2] (see 5.5.3.1.2). The PAN, <PAN>, and the Horizon_Txn_Num, <HTxnNum>, are also required to be passed in. This call is synchronous, and is hardware-based.
- The encrypted value (binary) is converted to Base64-encoding and inserted as the value of the [R2] < PIN_n > attribute.

Failures cause a Failure-[A3] message to be returned to the Counter.

5.1.3.4.5 Signing the [A3]

The Digital_Signature attribute, <DSig>, signs the <A3Mess> compound attribute in its RAG form. The processing steps to construct it are as follows. *(Unchanged from the S70 version of the NBS Authorisation Agent, including the PKC compression.)*

- The RAG form of <A3Mess:> is passed to the crypto subsystem (**crySignDataEx** function) to generate the binary Digital Signature (see 5.5.3.2.2). The signing is software-based and is mill-intensive.
- The binary Digital Signature is then Base64-encoded (**cryBinTo64** function) and inserted as the value of the <DSig> attribute.

With ICC Transactions (i.e. those with Entry_Method “ICC PIN Pad”), the extra payload of ICC Data in the [A3] can result in the size of the [A3] exceeding the Riposte limit of 2Kb. A substantial proportion of the [A3] data is the in-line Public Key Certificate (PKC) in the Digital Signature. The **crySignDataEx** function can be requested to return a binary Digital Signature with the PKC compressed sufficiently to prevent the [A3] from overflowing. Note that, once upgraded to S70, the Counter-based crypto subsystem will be able to recognise a compressed PKC and expand it before use, with no change to the Counter application (there may be some pre-S70 Counters in the field even after the deployment of the S75 Agent).

In this version of the Agent, compression of the PKC is ‘forced’ only for Transactions with Entry_Method “ICC PIN Pad” (value 3) or “ICC swipe fallback” (value 4). This is safe, as only upgraded Counters can generate such transactions. For other transactions, the parameter is set to ‘do not compress’.

Signing failures cause an unsigned [A4] message to be written in place of the [A3]. An [A4] is a variant of an [A3] that is not returned to the Counter, and is for diagnostic purposes only.

5.1.3.4.6 Use of AWKs

A session with an FI_EE’s PI cannot be used to transmit business transactions until an Acquirer Working Key (AWK) has been agreed for that session.

When an AWK is transmitted between the Agent and the PI, it is protected by an Acquirer Zone Master Key (AZMK). This is changed every six months or so. The operator Change_AWK command supports testing of new AZMKs when they change. The Agent does not need to provide any other support for changing AZMKs.

Further information is given in 5.1.3.2.3 and 5.5.3.1 and in the Key Management Design [KMSHLD].

5.1.3.5 Structure, Launch and Concurrency

The NBX Authorisation Agents are modelled on the high throughput and high availability Enquiry Agents developed for the NBS, DCS and ETS Authorisation Agents

The Agent runs as an NT Service, and is launched and relaunched by Tivoli. It will be run under its own Service User name (see Table 30).

A separate instance will be required to run against each Logical FI.

A single instance of the Agent will be capable of supporting the entire workload for the Logical FI. However, a second instance will run, typically at the other Campus, to act as a hot **Standby** (see 5.1.3.8). In this document, an Agent instance regards the other instance in such a **Resilient Agent Pair** as its **Partner**.

5.1.3.5.1 Threads for Business Functionality

In order to handle the required throughput the agent will include at least the following threads for the business functionality.

- A pool of **GetR1 Threads**, one per logical Routing Agent, to receive [R1]s and [C0]s and to return [A3]s (and exceptionally [A4]s)
- A pool of **Verify Threads** to verify Digital Signatures on the [R1]s and [C0]s
- A pool of **PreEE Threads** to format [R3]s and [E1]s
- A pool of **PI Handler** (or **EE_I/O**) **Threads** to communicate with FI_EEs, one per PI Handler
- A pool of **FIMngt Threads**, one per PI Handler, to handle Network Management messages
- A pool of **PostEE Threads** to process [A2]s and [E2]s
- A pool of **SignA3 Threads** to format and sign [A3]s
- A **Guaranteed Reversals Thread** to poll the C0 Reversals table for [C0]s
- A **Reversals Management** (or **E1Retry**) **Thread** to handle the must-deliver aspects of Reversals
- An **Exceptions Thread** to handle abnormal situations on behalf of (chiefly) the PreEE and PostEE Threads
- An **EE_Probe Thread** to handle health probes for inward connections
- A **Commands Thread** to poll the NPS Commands table

These threads are *persistent*. Apart from the inherent performance benefit, this is also exploited by the caching within the Crypto API DLL.

In addition to these business-related threads, threads are needed for control purposes, principally the Heartbeat Thread.

5.1.3.5.2 Heartbeat Thread

A control thread, known as the **Heartbeat Thread**¹⁰, is responsible for various miscellaneous tasks that are required for controlling the Agent and for monitoring the resources upon which the Agent is dependent. In effect, it manages all aspects of resilience, including:

- generating Heartbeats in the NPS indicating the health of the Agent to its Partner
- generating Heartbeat History information
- recording statistical information (see 6.2.2.2)
- monitoring the Heartbeats from its Partner, and making decisions on failover
- monitoring the connection to NPS, and switching between NPS Oracle instances
- monitoring the Administrative Status of the FI_EE
- monitoring other necessary resources
- monitoring the time difference between its clock and the NPS's

5.1.3.5.2.1 Heartbeats

The exchange of Heartbeats between an Agent and its Partner is used:

- to vote as to which of the two Agent instances is the **Active Agent** and which the **Standby**. Unlike previous Authorisation Agents, this voting only happens if both instances start at near enough the same time
- by the Active Agent to resign in favour of the Standby, prior to its failing
- as a source of status information for operational management of the Agent – the Heartbeat history is harvested to OMDB (see 6.2.1.2)

An important aspect of monitoring Heartbeats from one's Partner is to detect when the Partner has failed to refresh the Heartbeat within a reasonable period. The Standby Agent expects the Heartbeat to be refreshed on a regular basis.

This topic is discussed in 5.1.3.8.1.

¹⁰ There are actually two Heartbeat Threads, with the second monitoring the first, each connecting to a different NPS Oracle instance.

5.1.3.5.1.2 *Monitoring the connection to NPS*

As discussed in the section on resilience (5.1.3.8.2), in the event of a failure in the connection to an NPS Oracle instance, the Agent attempts to use an alternative Oracle instance.

There are two Heartbeat threads, each connected to a different Oracle instance. Each Heartbeat thread uses the reading of the Heartbeat table to monitor the health of the connection to its Oracle instance.

5.1.3.5.1.3 *Monitoring other necessary resources*

If the Agent is unable to offer an authorisation service because, for example, it is unable to access a necessary resource, the Agent will deem itself to be “**Unavailable**”. A Heartbeat message will inform its Partner of its unavailability.

The only resource-related case appears to be when the Agent is unable to access the required cryptography services or these services have reported an irrecoverable failure.

The Agent is also Unavailable if it is closing down, for whatever reason. Failover is quicker and cleaner if this fact can be passed to the Agent’s Partner, but it may not be possible in all failure scenarios.

The term “Unavailable” is restricted to those scenarios where Fujitsu Services should shoulder the blame for the inability to offer an authorisation service.

5.1.3.6 **Exception Handling**

Operational failures are treated by this Agent in a different way from traditional Harvester Agents. This is partly because of the requirements of high availability, which has led to the concept of a Resilient Agent Pair and of failover to a Standby Agent (see the section on Resilience, 5.1.3.8).

The difference in approach is also because of the need to apportion blame for a failure to the offending party. If, for example, an FI_EE or one of its PIs fails, the Agent should not fail because it has lost its connection – it needs to continue to operate and generate Failure-[A3]s with a Response_Code that clearly assigns the blame to the FI_EE.

Note that all failures and exceptions are recorded in the NT event log.

The treatment of validation failures, and of the timeout of an [A1] response from the FI_EE, is covered in detail under the Functional Description above.

A call to the Generate Digital Signature function (see 5.5.3.2.2) will return a failure condition only following a serious operational failure in the crypto subsystem. The Agent will treat this as a loss of a dependent resource, and will accordingly fail.

On the other hand, a failure with verifying a Digital Signature or decrypting the Sensitive Data is likely to be specific to one Outlet. The Agent will attempt to categorise the failure responses into those that should allow the Agent to continue and those that represent the loss of a dependent resource and will accordingly fail.

A call to the Translate PIN Block function (see 5.5.3.1.2) will return a failure condition either if the MAC check fails (which indicates data corruption or a possible security attack) or following a serious operational failure in the crypto subsystem. With the former the failure is likely to be specific to one Outlet and so the Agent will be allowed to continue; but the latter represents the loss of a dependent resource and the Agent will fail.

PIN Block translation is dependent upon crypto’s Load Balancing service. Should this service stop for any reason, experiment has shown that the next attempt to translate a PIN block will hang indefinitely. Certain error conditions with the crypto hardware can cause a similar hang. The Agent has to guard against this. It will need to divide the work between two threads, one making the call, the other monitoring that the call is not taking an unduly long time to return. If it detects that it has taken longer than a configurable threshold, it will

cause the Agent to fail. The same considerations apply to the NbxInitialiseCrypto function (see 5.5.3.1).

5.1.3.7 Performance and Scalability

5.1.3.7.1 Targets

[BUSVOLS] provides NBX business volumes for the Authorisation Agent. The volumes are expressed as the number of transactions per second for the peak five-minute period of the month. The **Contracted Volume** figure is the maximum volume that Fujitsu Services will contract to support. The **Design Limit** figure is the volume that the system will support without significant failures.

Average transaction rate in peak 5-minute period	Contracted Volume	Design Limit	Comment
CAPO	160	192	From [BUSVOLS]
LINK	32	39	From [BUSVOLS]
A&L	12	14	From [BUSVOLS]
Total	204	245	Total volumes from [BUSVOLS]
Per CAPO Agent instance	84	101	Allow 5% skew between CAPO_A and CAPO_B

Table 8 – Authorisation Agent Performance Targets (transactions/sec)

Note that these are the number of [R1] messages to be processed, to which must be added the Reversals. The normal proportion of Reversals is low, maybe 2%, but under periods of overload where stale [R1]s have to be discarded, the number of Reversals through the 'real-time' route can be as many again. If this coincides with an NBX Guaranteed Reversals Agent delivering a large number of duplicates following a failure, the backlog of 'guaranteed' Reversals can become very large indeed.

5.1.3.7.2 Analysis

The Authorisation Agents must be designed so that they not only successfully process the Contracted Volumes, they must also meet the Design Limit volumes 'without significant failure'. In other words, they must remain stable under stress when overloaded. If necessary they can discard a proportion of messages to protect themselves, but even then they need to be able to successfully handle the Contracted Volumes.

The high volumes for CAPO require that the workload is split between two Agents: CAPO_A and CAPO_B. Furthermore, separate platforms are required for the two Agents in each Resilient Agent Pair, so a total of four platforms is required. On the other hand, the low volumes for LINK and A&L mean that only two platforms are required for each of these FIs.

5.1.3.7.2.1 Cryptographic overheads

The burden of the processing that the Verify, PreEE and SignA3 threads perform is the calls into the cryptographic subsystem. The figures are given in Table 9 in the order that they are performed.

Step	Thread	S/W ms at Phase 2	H/W ms
------	--------	-------------------	--------

Signature verification	Verify	7.5	
Decryption	PreEE	3	
PIN Block translation	PreEE		1.9
Signature generation	SignA3	5	

Table 9 – Cryptographic timings for processing an authorisation transaction

The NBX Authorisation Agent Server is a 4-processor platform. The high proportion of pure mill shows is there is likely to be little gain by increasing the number of the mill-intensive Verify and SignA3 threads. Indeed, a 'single-threading throttle' in the crypto code has been demonstrated to show that there is little benefit in having many such threads at all. In order to ensure that signing is given precedence over verifying when the Agent is under pressure, and to ensure that any queuing for a bottleneck is for verifying rather than for signing, the number of threads is currently set to 1 Verify and 3 SignA3 threads.

With this severe throttle in the Software crypto code, the Authorisation Agent for CAPO would be unable to meet the Contracted Volume. Therefore it was decided, via CP3896, to suppress the checking of Digital Signatures for [R1] messages already protected by having a (single) PIN block present. This suppression has been configured only for the CAPO Agent, as that is the only one where the Design Limit is high enough to warrant it. Suppression is further discussed in section 5.1.3.1.1.

Signature checking must not be suppressed for [C0] messages. This was one reason why it was decided to introduce the concept of 'staleness' for real-time [C0]s in the same way as it applies to [R1]s. If backlogs are building up within the Agent, this can apply back-pressure to the queue of messages waiting for their signatures to be verified. Therefore, a staleness check is made on real-time [C0]s immediately before the verification would be performed.

As staleness cannot and must not apply to guaranteed [C0]s, alternative techniques must be used to throttle them so that they do not overwhelm the signature verification bottleneck – see below.

Having by one means or another reduced the throughput requirements on signature verification, there is no need for the Agent to take specific measures for the other cryptographic functions.

5.1.3.7.2.2 NPS considerations

For most normal [R1] transactions, there are necessarily at least two interactions with the NPS:

- an audit (or journal entry) of the [R3] immediately before sending it to the FI_EE
- an audit (or journal entry) of the [A1] immediately after receiving it from the FI_EE

In addition, the transaction's status must be recorded for recovery purposes, and duplicate transactions must be checked for.

The Authorisation Agents are designed so that they make precisely two interactions in the normal case. The status and journal records are written and committed all in a single interaction. To prevent unnecessary reading of the status records, any update of the status record is qualified by a 'WHERE' clause checking that the transaction's current status is as expected. Only if an insert does not succeed as expected, or the update does not succeed because the current status is not as expected, is the status record explicitly read and the appropriate actions taken.

Indexes in NPS

The topic of partitioning the tables and maintaining indexes on key fields is fully described in the NPS High level Design, [NBSHLD]. Suffice it to say here that:

- There are separate tables for each Logical FI for the most critical tables, including the Transaction Status, Transaction Journal and Heartbeat tables.
- There is now no index on the 'current status' in the Transaction Status table. It was expensive to maintain, and the benefit only accrued on an Agent restart when searching for 'must-deliver' Reversals that had not yet been delivered.

Interactions with TES

As described in 5.1.3.1.6.2, entries written to the Transaction Journal are organised in 'batches'. The TES extraction process is harvesting these batches as soon as the batch is complete. There is one such TES-CO harvester for each Logical FI.

To optimise the performance it is highly desirable, though not absolutely necessary, for the TES extraction process to harvest from the same NPS node as the Authorisation Agent has written to. In normal running, this is achieved simply by matching configuration of both components: CAPO_A, LINK_A and AL_A use NPS1, and CAPO_B, LINK_B and AL_B use NPS2.

If an Authorisation Agent has trouble with the connections to its preferred NPS node, it switches to the other node. The chances are that the TES-CO harvester will also have trouble with its preferred node, so likewise will switch to the other node. On the other hand, it may be that on occasions they are running using opposite nodes for the rest of the day – the NPS can handle this.

Every night, the TES-CO harvester is stopped for overnight processing and the new instance will revert to its preferred NPS node. Therefore, the Authorisation Agent needs to 'rehome' to its preferred NPS node during the night, but this does not have to be specifically co-ordinated with the TES-CO harvester (see 5.1.3.8.3).

NPS as a bottleneck

Performance testing on the Volume & Integrity rig showed, unexpectedly, that the interactions with the NPS could be subject to variable and sometimes severe delays in periods when busy. (A programme of improvements was undertaken throughout most of 2005, but these are outside the scope of this HLD.)

These delays would lead to other [R1]s becoming stale as they waited queued behind affected transactions. The original design was to journalise the stale [R1]. The transaction would time out at the Counter, resulting in a [C0] and a further journal record. In other words, this still resulted in two interactions with the NPS. As it was the NPS that proved to be the bottleneck the journalising of stale [R1]s had to be abandoned, despite the specific requirement (NBR0178) to journalise them.

The potential NPS delays also required that the threshold used for the staleness check had to be much more aggressive. Initially, it had been set to 30 seconds, only a little less than the timeout at the Counter. This was reduced to 15 seconds. This is so that the heavy resource used to verify the digital signature is not wasted by the transaction eventually timing out anyway because of NPS delays. It is also so that any [A1]-Accept is passed back to the Counter in good time even if there is an NPS delay in journalising the [A1].

5.1.3.7.2.3 *Guaranteed Reversals*

In normal running the Authorisation Agent's 'Grev' thread, which provides the sole interface to the C0 Reversals table, should not find a large number of C0 Reversals that require processing. To optimise the Select statement, the Agent maintains in memory a 'high-water mark' of Reversals that have been previously selected, using the record's sequence number that had been generated by the NBX Guaranteed Reversals Agent from an Oracle sequence. (Actually, this high-water mark and the sequence numbers are per Riposte Cluster.)

However, in a recovery situation where there may be a large number of unprocessed Reversals in the C0 Reversals table, it is necessary:

- to avoid the overhead of unnecessary Select statements (especially given the ordering that has to be performed); and
- to prevent the Reversals from swamping the rest of the Agent.

This is achieved:

- by placing all Reversals that have been selected and fetched on an internal holding queue;
- by severely restricting the number of Reversals that can be released into the rest of the Agent at any one time (to 100, say); and
- by imposing a slight delay (10 msec, say) between every such release.

Note that if the 'real-time' version of the C0 has already been captured by the Agent, the 'guaranteed' Reversal is immediately marked by the Grev thread as 'processed' in the C0 Reversals table and so does not contribute to the above workload.

5.1.3.8 Resilience

Mechanisms are required to ensure that, should an instance of the Agent fail, another one will take its place. The case of an NPS Oracle instance failure (including a failure of the network linking the Agent Server to the Oracle instance) also needs to be considered. In this case, the Agent process will continue operating, but use the other local Oracle instance.

In the case that an Agent instance fails, then all outstanding transactions being processed by that Agent will be abandoned (and will eventually timeout at the Counter). No attempt will be made by the replacement Agent to recover such work.

5.1.3.8.1 Agent Failover

Resilience to Agent failure is achieved by having two instances of the Agent running at the same time on different platforms (typically at different Campuses), one running as the Active Agent, the other running as a Standby. When an Agent instance first starts up, it runs as a Standby.

If the Active Agent fails, the Standby Agent takes over and become the Active one. For this approach to be viable, a reliable communications link is required between each instance of the Agent. This is provided by a heartbeat table in the NPS.

5.1.3.8.1.1 Heartbeats

Each Agent instance will refresh its Heartbeat in the NPS every five seconds (configurable in registry). Each of an instance's Heartbeats overwrites its previous Heartbeat, so only its most recent Heartbeat is available. *(Note that the separate Heartbeat History table makes Heartbeat information available for systems management purposes – see 6.2.1.2.)*

The Heartbeat includes the following information for failover purposes:

- *Agent_Active* – Whether the Agent instance is actively processing NBX transactions. Values of 'Y' and 'N' are used
- *Agent_Unavailable* – Whether the Agent instance is operational, i.e. whether it would be able offer an authorisation service at all. Note that a standby instance is normally operationally available. Values of 'Y' and 'N' are used
- *EE_Bad* – The 'availability' (see 5.1.3.2.4) of the Agent instance's connections to the FI_EE's PIs. Values of 'Y' and 'N' (meaning Down and Up respectively) will be used. If only some PIs are Down, then a composite value is used. Only relevant for an Active instance (as a Standby instance does not establish any connections)
- *Priority* – A static priority value obtained from the NBX Authorisation Agent Server registry. This is used to determine which Agent instance has priority at Agent service

start-up. It is also used to determine which Agent should 'resign' when they are both Active. The values 1 & 2 should be used, with 1 being the higher priority

As Heartbeats are also used for operational monitoring of the Authorisation Agent instances, an Agent instance will write a Heartbeat even if it is operationally unavailable.

Each Agent instance will connect to both NPS Oracle instances, so that it can read Heartbeats from both. (Note that they are written to only one Oracle instance.) Failure of either connection will require the Agent to enter a retry loop until connection can be re-established. Failure of both connections will result in the Agent exiting, at which point it will be restarted by Tivoli/EACRR.

Each Agent instance will attempt to write a final Heartbeat when it closes down, in both normal and exception scenarios, to inform its Partner what it is doing. *Agent_Active* will be set to 'N' and *Agent_Unavailable* to 'Y'.

There are two distinct failover scenarios to be considered:

- Both Agent instances are running and can control the decision
- Expected Heartbeat refresh by the Active Agent has not happened

The LLD, [NBXAUTH], gives a more detailed analysis of the Agent's behaviour. This HLD captures the main points.

5.1.3.8.1.2 *Controlled failover*

Upon reading a Heartbeat, an Agent instance has to decide whether it is the one that should be the Active instance.

Controlled failover from an Active instance to a Standby instance occurs only when the Active instance 'resigns' following loss of a critical resource. In this case, its final Heartbeat will set *Agent_Unavailable* to Y. (Note that an Active instance will not resign if its Partner is Unavailable.)

If more than one instance is Available, the decision-making between them is as follows:

- *Agent_Active* – an Active instance takes precedence over a Standby instance (i.e. a standby instance will never negotiate to wrest control from an Active instance)
- *Priority* – Everything else being equal, the instance with the lower priority number will take precedence¹¹. This is relevant if both instances are Standby's, as could have if both are started near enough the same time, or both are Active (it is believed that this cannot happen)

5.1.3.8.1.3 *Unilateral takeover*

A Standby instance will take over when it detects the non-refresh of the Active instance's Heartbeat for 25 seconds (configurable via registry).¹²

5.1.3.8.2 NPS Failure

The NPS will be accessible via two different Oracle instances. Each NBX Authorisation Agent Server will be physically connected to both Oracle instances such that an NBX Authorisation Agent should at all times be capable of connecting to one or other of them.

When a connection to one Oracle instance fails, any NBX Authorisation Agents connected to it need to automatically fail over to use the other Oracle instance. Failover should be achieved within a few seconds, therefore minimising the period that the authorisation service is affected. This failover will be achieved within the Agent application.

¹¹ Priorities will only be equal in a misconfigured system. To allow for this, the Agents will use a final discriminator, such as alphabetical order of host name.

¹² The algorithms for both controlled failover and unilateral takeover will take account of the exceptional running of three or more Agent instances.

Should the Agent lose its connection to both Oracle instances, it will fail.

During the connection phase when the Agent is first loaded, the Agent attempts to connect to both Oracle instances. Until it has succeeded connecting to at least one of them, it adopts the standard Agent approach of retries as appropriate; retries continue until the configured total_connection_timeout period has elapsed, after which the Agent fails.

Once it has connected to one Oracle instance, the Agent can enter its main processing phase. Attempts to connect to the other Oracle instance continue indefinitely (and are not controlled by the total_connection_timeout).

Similarly, following a failover from one Oracle instance to another, attempts to re-establish resilience are made by trying indefinitely to reconnect to the original Oracle instance.

5.1.3.8.3 Rehoming the Agent

For efficiency, the Authorisation Agent should write its Transaction Journal records to the same Oracle instance as the TES collection process is harvesting the records from. This is material during the core day, merely desirable outside core hours. Basic configuration of the both Agents and TES will ensure that their preferences for which Oracle instance to use match. They can get out of step as a consequence of some form of failover by Agent or by TES.

Each night, TES processing is closed down for backup, with the effect that the TES processes will revert to connecting to their preferred Oracle instance. Therefore, the Agent also needs to revert each night to using its preferred Oracle instance.

As an independent requirement, to aid operations staff predict where active Agent instances will be running, an active Agent running on its secondary server will 'resign' in favour of the standby Agent running on the primary server. The primary server is the one with the one with higher priority (i.e. the lower numerically, see 5.1.3.8.1.1).

The Agent performs these two rehoming tasks – the active connection reverting to the preferred Oracle instance, and the active Agent reverting to its primary server – together at 02:15am (configurable by registry). The time is only loosely coupled to the TES overnight processes, as the workload during the night is minimal. Rehoming the Oracle instance is done by both active and standby Agents. But only the active Agent is involved in the decision to rehome to the primary server.

5.1.3.8.4 Network Failure

The networks between the NBX Authorisation Agents and the FIs have been designed to be highly resilient. As such, an active Agent will not fail over to its standby in the event of network problems – the standby Agent is very unlikely to find the network in better shape.

5.1.3.8.5 FI_EE failure

The FIs are outside the Horizon operational domain and therefore the Agent will not fail on failure of connections to the FI_EEs.

The NBX Authorisation Agent will respond to [R]s that do not receive an [A1] within a set timeout period. This mechanism will be used to calculate the service levels achieved within the Horizon domain. Failing the Agent over following FI_EE failure would result, for the period of the failover, in Fujitsu Services being penalised for the failure of a component outside its operational domain.

5.1.3.9 Configurability

5.1.3.9.1 NBX Configuration Parameters table in NPS

Section 5.1.1.4.2 has given an overview of the NBX Configuration Parameters table (the TMS_TX_NBX_CONFIGURATION table) in NPS.

[BUSPARAMS] documents those 'business parameters' that are subject to some level of change control between Post Office Ltd. and Fujitsu Services Ltd.

This section defines those NBX Configuration parameters that that Fujitsu Services Ltd. can change without prior consultation with Post Office Ltd.

5.1.3.9.1.1 *Transaction Journal control parameters*

PARAMETER_NAME	APPLIES_TO	PARAMETER_VALUE	DESCRIPTION
Jnl	Description		Control parameters for writing to the Transaction Journal in NPS
Jnl.Partition.MaxSz	Description		Approximate maximum number of journal entries in a partition
Jnl.Partition.MaxSz	Generic	40000	
Jnl.Batch.MaxSz	Description		Approximate maximum number of journal entries in a 'batch' (logical subpartition)
Jnl.Batch.MaxSz	Generic	10000	
Jnl.Batch.Interval	Description		Approximate interval (milliseconds) before switching to a new 'batch'
Jnl.Batch.Interval	Generic	30000	30 seconds
Jnl.Batch.MaxInterval	Description		Maximum interval (milliseconds) before switching to a new 'batch'
Jnl.Batch.MaxInterval	Generic	600000	10 minutes
Jnl.Force.Delay	Description		When an Agent has failed it is not in a position to make its last batch Available. After 'force delay' (msecs), the newly active Agent will make Available any alien batch. Must be greater than HB_CONTROL.HB_MAX_GAP registry. NULL implies facility is off.
Jnl.Force.Delay	Generic	120000	2 minutes

Table 10 – Transaction Journal control parameters

5.1.3.9.1.2 *STAN-generation control parameters*

PARAMETER_NAME	APPLIES_TO	PARAMETER_VALUE	DESCRIPTION
STAN	Description		Control parameters for generating Systems Trace Audit Number (bitref. 011)
STAN.Max	Description		Maximum value of STAN
STAN.Max	Generic	999999	
STAN.Margin	Description		Distance of threshold from batch end - i.e. when a STAN is within STAN.Margin of the end of the batch, it is time to request a new batch
STAN.Margin	Generic	2000	
STAN.Thread.Wait	Description		Approximate interval (milliseconds) allowed for thread to reserve a new batch of STANs, before another thread can try
STAN.Thread.Wait	Generic	15000	15 seconds
STAN.Batch	Description		Number of STANs to reserve in a batch
STAN.Batch	Generic	10000	

Table 11 – STAN-generation control parameters

5.1.3.9.2 Registry

Table 13 lists the more important items that are configurable through the Registry.

☞ The default values given in this table are indicative only. The reader should consult the Low Level Design document [NBXAUTH] for authoritative default values and for the names of the Registry values.

Item	Description	Section	Default
Agent identity (IDENTITY)	Identifies the Logical FI the Agent services. One of "CAPO_A", "CAPO_B", "LINK_A", "LINK_B", "AL_A", "AL_B"		–
FI_EE's host names (SOCKET_HOST)	Used to look up the IP addresses. The host names must follow a particular naming convention		–
FI_EE's service names (SOCKET_SERVICE)	Used to look up the port numbers. The service names must follow a particular naming convention		–
Number of virtual addresses per PI Handler (VA_COUNT)	CAPO: 2, one per remote Thread LINK and A&L: 1	5.1.2.3.1	CAPO: 2 else 1
Number of sockets (i.e. TCP/IP connections) per PI Handler (SocketConcurrency)	CAPO: 2, one per remote Thread LINK and A&L: 1 The same values must be used for SocketConcurrency and VA_COUNT to ensure there will be one connection per virtual address		CAPO: 2 else 1
Wait for stale connection to die (WaitToDie)	Period to wait for an existing inward connection to die before rejecting a new inward connection (only for LINK; from S90)	5.1.3.2.2.2	40 secs
Allowance before replacing closed socket	Minimum delay before attempting to create a replacement connection following the failure on a socket		2 secs
Maximum timeout on the FI_EE	Value above which the timing out of the [A3] response is not guaranteed to behave correctly		30 secs
Flag to suppress Signature verification	Applies only to [R1]s with a single PIN block	5.1.3.1.1	N
Number of threads in each pool			

Quick wait	Duration of short sleep when FI_EE I/O Handler has nothing to send to the PI but responses are expected.	5.1.3.7.2.3	20 msec
	Duration of short wait between each Reversal released by the Grev thread to the rest of the Agent		10 msec
Wait sleep	Duration of short sleep when FI_EE I/O Handler has nothing to send to the PI and no responses are outstanding		200 msec
Crypto timeout	Time after which a call to the NB Crypto DLL is deemed to have hung		15 secs
Controlled takeover delay	Minimum interval after an Active Agent has stood down before it will become Active again by controlled takeover		3 mins
Priority	Different priorities need to be assigned to each Agent instance		1, 2
Statistics interval	Frequency of writing Statistics records		1 min

Table 12 – Configuration of NBX Authorisation Agents

5.1.3.10 **Audit**

Messages sent to and received from the FI_EE cross a Service Boundary and therefore are required to be audited. To achieve this, they are written to NBX Transaction Journal in the NPS. The exception is the Handshake messages.

The messages with CAPO include binary rather than hexadecimal character representations of the Primary and Secondary Bitmap fields. These are audited as hexadecimal characters, as the audit subsystem handle binary data.

Sensitive data is obliterated before it is journalised.

5.1.3.11 **Operational Summary**

5.1.3.11.1 NBX CAPO Authorisation Agent

Agent name: NX_NQ_CAPO	Platform(s): NBX Authorisation Agent Server
Service Name: TMSNX_</I>_<s>	Style: Resilient Enquiry Agent, with hot standby
Scope & parallelism: One per Logical FI per Site (</I> is “CAPO_A” or “CAPO_B”, <s> is the site); multithreaded. One of the pair for each Logical FI is acting as the hot standby for the other.	
Registry key(s): HKEY_LOCAL_MACHINE\SOFTWARE\NCL\PathwayAgents\NX_NQ_CAPO HKEY_LOCAL_MACHINE\SOFTWARE\NCL\PathwayAgents\NX_NQ_CAPO\TMSNX_</I>_<s>	

Use of checkpoints: None.
Use of dummy offices: None.
Host database: NPS
Needs to be running: Runs 7 x 24. Important between 07:00 and 20:00 7 days per week; Critical between 08:00 and 17:30 Monday to Friday and 08:00 and 13:00 Saturday.
Documentation: [NBXAUTH]

5.1.3.1.2 NBX LINK Authorisation Agent

Agent name: NX_NQ_LINK	Platform(s): NBX Authorisation Agent Server
Service Name: TMSNX_</I>_<s>	Style: Resilient Enquiry Agent, with hot standby
Scope & parallelism: One per Logical FI per Site (</I> is “ LINK_A ” or “ LINK_B ”, <s> is the site); multithreaded. One of the pair for each Logical FI is acting as the hot standby for the other.	
Registry key(s): HKEY_LOCAL_MACHINE\SOFTWARE\ICL\PathwayAgents\NX_NQ_LINK HKEY_LOCAL_MACHINE\SOFTWARE\ICL\PathwayAgents\NX_NQ_LINK\TMSNX_</I>_<s>	
Use of checkpoints: None.	
Use of dummy offices: None.	
Host database: NPS	
Needs to be running: Runs 7 x 24. Important between 07:00 and 20:00 7 days per week; Critical between 08:00 and 17:30 Monday to Friday and 08:00 and 13:00 Saturday.	
Documentation: [NBXAUTH]	

5.1.3.1.3 NBX A&L Authorisation Agent

Agent name: NX_NQ_AL	Platform(s): NBX Authorisation Agent Server
Service Name: TMSNX_</I>_<s>	Style: Resilient Enquiry Agent, with hot standby
Scope & parallelism: One per Logical FI per Site (</I> is “ AL_A ” or “ AL_B ”, <s> is the site); multithreaded. One of the pair for each Logical FI is acting as the hot standby for the other.	
Registry key(s): HKEY_LOCAL_MACHINE\SOFTWARE\ICL\PathwayAgents\NX_NQ_AL HKEY_LOCAL_MACHINE\SOFTWARE\ICL\PathwayAgents\NX_NQ_AL\TMSNX_</I>_<s>	
Use of checkpoints: None.	
Use of dummy offices: None.	
Host database: NPS	
Needs to be running: Runs 7 x 24. Important between 07:00 and 20:00 7 days per week; Critical between 08:00 and 17:30 Monday to Friday and 08:00 and 13:00 Saturday.	
Documentation: [NBXAUTH]	

5.1.4 NBX Guaranteed Reversals Agent (NX_HV_GREV)

5.1.4.1 Overview

To ensure that every Counter-generated [C0] Reversal is delivered, if appropriate, to the FI, it is necessary to provide a reliable route for transferring [C0]s from the Counter to the NBX Authorisation Agent.

The **NBX Guaranteed Reversals Agent** provides this assurance. It will harvest all [C0] messages from the Correspondence Servers using a checkpointed message port to ensure that every [C0] is eventually harvested. It will harvest them into the **C0 Reversals table** in the NBX Persistent Store (NPS), from where the NBX Authorisation Agent will pick them up and process them. Only the NBX Guaranteed Reversals Agent populates the C0 Reversals table.

The NBX Guaranteed Reversals Agent does not check the digital signatures on the [C0]s – this is left to the NBX Authorisation Agent.

If the NBX Authorisation Agent has already reversed the Transaction, because, for example, the [C0] has already been received directly, the duplicate will not be processed. However, if the [C0] that is processed has arrived though this guaranteed route, the record in the Transaction Journal will be marked accordingly, so that the efficacy of the route can be monitored (but only on an *ad hoc* basis).

The NBX Guaranteed Reversals Agents could harvest the same [C0] more than once. The NBX Authorisation Agent will ensure that duplicates are ignored.

5.1.4.2 Structure, Launch and Concurrency

The NBX Guaranteed Reversals Agent is a database-coordinated Interactive Harvester. There will be one Agent instance per Riposte Cluster. It runs on an NBX Routing Agent Server platform.

The Agent runs as an NT Service, and is launched and relaunched by Tivoli. It will be run under its own Service User name (see Table 30). It will not be dependent on any Correspondence Server in a different Cluster, nor will it be dependent on the Cluster Lookup Service.

The pulse_interval used by the Agent needs to be different from those used by other Agents, so that they don't synchronise their checkpointing. The value chosen will be similar to that used by the NBS Expedited Confirmation Harvester, which it supersedes, but as there will be a brief period of parallel running the value will not be identical.

The parameter controlling checkpointing, namely the pulse_interval, should be configured so that following a failure any outstanding work can be caught up within five minutes of the agent restarting. Checkpoints are divorced from commits to the database, which is controlled by the success_unit_count parameter. The proposed values are:

- pulse_interval: 1,175,000
- success_unit_count: 50

Restarting from a checkpoint necessarily means that [C0] messages can be passed to the C0 Reversals table more than once. The handling of such duplicates is handled correctly by the NBX Authorisation Agents (see 5.1.3.1.3).

5.1.4.3 Detailed Processing

Every Network Banking [C0] message is read and processed. The filter on the message port is therefore:

- <Data.Ctrl.MsgType:C0>
- <Application:NBA>

The Routing_Gateway and Agent_Hash attributes in the [C0], together with the number of the Cluster the Agent is servicing, are used to determine which Logical FI is to process the message. Accessing and using the NBX Routing Data is the same as that for the NBX Routing Agent – see 5.1.1.4.1 and 5.5.1.2.

There is little other validation that can be performed. Specifically, the Digital Signature is not verified and the [C0] is written to the C0 Reversals table as is. The Agent extracts from the [C0] the data that forms the key that will be used by the NBX Authorisation Agent to access the Transaction Status table, and writes this along with the [C0] itself.

Every [C0] written to the C0 Reversals table is assigned a monotonically increasing sequence number, as an aid to the efficient reading of 'unactioned' [C0]s by the NBX Authorisation Agents. An Oracle Sequence per NBX Guaranteed Reversals Agent (i.e. per Riposte Cluster) is used to maintain four separate series of [C0]s.

5.1.4.4 Exception Handling

The Agent will treat operational failures in the same manner as do other Harvester Agents. Note that all failures are recorded in the NT event log.

As part of its initialisation, the Agent reads the NBX Routing Data. The Agent performs the same consistency checks on the Routing Data as does the NBX Routing Agent – see 5.1.2.3.2. If the data is faulty the Agent will exit with an appropriate exception value.

If a message cannot be routed because there is no matching Routing Data, the behaviour depends upon the state of a registry switch. During the period of migration, when the NBX system is only processing specific Routing Gateway values and the NBS Expedited Confirmation Agent is still routing [C0]s to NBE, the NBX Guaranteed Reversals Agent will silently ignore the message and leave it to NBE to process appropriately. Once the NBE has been switched off, the registry switch will be thrown and the unroutable [C0] will be treated as an exception. It will be written to a **C0 Exceptions Table** in the NPS, and reported in the NT Event Log. Measures will be taken to prevent an event storm in the event of a large number of exceptions.

Serious faults in the [C0] message read from the message port, in particular mandatory attributes needed for routing or for the C0 Reversal Table are absent or invalid, will also result in the message being treated as an exception.

5.1.4.5 Performance and Scalability

It is expected that normally up to 2% of [R] messages will result in a [C0] being generated, so the workload on average will be up to 2% of that for the NBX Authorisation Agent. However following a failure of an Agent, a Correspondence Server or an FI-EE, then there could be peaks where 100% of [R]s will result in a [C0] being generated.

A failure could result in an NBX Guaranteed Reversals Agent instance having to reprocess all the [C0]s since a previous checkpoint.

It is important that the NBX Routing Agent instance(s) on the same platform is not starved of processor resource. Therefore, it is proposed that these Agents will be bound to one of the two processors on the platform. This restricts the maximum usage by this Agent to 50% of the platform processor resource.

*This is done by the Agent using a "SetProcessAffinityMask()" call to bind it to the first processor, i.e. the one defined by the 2**0 bit in the affinity mask. Note that which processor it binds to is not currently configurable.*

5.1.4.6 Resilience

Mechanisms are required to ensure that, should an instance of the Agent fail another one will take its place.

In this case, the Tivoli/EACRR mechanisms are considered to be adequate. Should an Agent instance lose connection with Riposte or the Oracle database, it fails and allows a new instance (on either the same or another platform) to try and re-establish connections. The aim of EACRR is to restart a failed Agent instance within 5 minutes or so; the Agent instance then has to repeat work from the previous checkpoint in order to catch up.

Such an approach is considered to be adequate because the NBX Authorisation Agent captures in real-time virtually every [C0] replicated from Counter to data centre. For such [C0]s the NBX Guaranteed Reversals Agent is merely providing a fail-safe backup route. On the other hand, for [C0]s whose replication has already been delayed, a further delay of up to 15 minutes engendered by restarting from a checkpoint is not really material.

Following a failure during the connection phase, the standard Agent approach of retries is appropriate; retries continue until the configured TOTAL_CONNECTION_TIMEOUT period has elapsed, after which the Agent fails. A timeout of the order of 5 minutes will be required so that EACRR may restart it on another NBX Routing Agent Server well within the 15-minute period.

The NBX Guaranteed Reversals Agent will be configured with a Resilient Locale. If the Riposte connection to a Correspondence Server fails, this Agent will fail and wait to be restarted by Tivoli/EACRR mechanisms. During the connection phase, it will attempt to connect to the preferred Correspondence Server, but if that is not possible it will connect to the alternative instead.

For connection to the NPS, the NBX Guaranteed Reversals Agent is configured with just one name, e.g. "NPS1". If it fails to connect within the TOTAL_CONNECTION_TIMEOUT period, the Agent fails and EACRR will restart it (at the second attempt at worst) on an Agent Server at the other data centre. Bootle and Wigan instances of the Agent are configured to connect to different NPS nodes, so this achieves failover from one NPS node to the other within the ORAC cluster.

Behind the scenes, as it were, the single name "NPS1" is actually a list of node 1 at the NPS primary site, Bootle, followed by node 1 at the disaster recovery site, Wigan, only one of which can be available at any one time; and similarly for "NPS2". The SQL client attempts to connect to each node of the list in turn. *(It was originally envisaged that the two-node list would be Bootle nodes 1 and 2. Then a four-node list was postulated, but this was found to be too long and caused timeout problems within the SQL client.)* During such a failover, the Agent has to guard against using sequence numbers out of order. Without taking special precautions, the caching by Oracle of a batch of values for a Sequence could result in numbers being used out of order.

5.1.4.7 Configurability

NBX Routing Data is described in section 5.5.1.2.

Table 13 lists the more important items that are configurable through the Registry.

Item	Description	Section	Default
Cluster number (CLUSTERID)	Number of the Riposte Cluster for this Agent instance		
Resilient Locale	Name of a Resilient Locale configured in the CLUSTER_LOOKUP_SERVER registry	As for NBS – see [NBSHLD]	
Database (DBLOCATION)	The name of an entry in the tnsnames.ora file, typically "NPS1" for Bootle Agents and "NPS2" for Wigan Agents	5.1.4.6	

Dummy office for configuration data	GroupId of dummy office used to hold Class D Reference Data used to configure Response_Code handling. 999961 to 999964 for Clusters 1 to 4 respectively		
Exception Rules	Control of how to handle exceptions. During migration, the default will be to ignore unroutable [C0]s	5.1.4.4	

Table 13 – Registry for NBX Guaranteed Reversals Agent

5.1.1.8 Audit

As the NBX Guaranteed Reversal Agent is not servicing an external interface there is no requirement for auditing the [C0] messages being processed.

Any [C0] messages to be written to the C0 Reversals table will be timestamped with the time that they are inserted into the table.

5.1.1.9 Operational Summary

Agent name: NX_HV_GREV	Platform(s): NBX Routing Agent Server
Service Name: TMSNXGRev<n>	Style: Database-Coordinated Interactive Harvester
Scope & parallelism: One per Cluster (<n> is the Cluster id)	
Registry key(s): HKEY_LOCAL_MACHINE\SOFTWARE\ICL\PathwayAgents\NX_HV_GREV HKEY_LOCAL_MACHINE\SOFTWARE\ICL\PathwayAgents\NX_HV_GREV\TMSNXGRev<n>	
Use of checkpoints: Starts from named checkpoint, error if it does not exist. Checkpoint name(s): AGT_NX_HV_GREV_<n>	
Use of dummy offices: The Agent Data Office (999993) for holding management information on the history of backup checkpoints; the Agent Configuration Office (99996<n>) for Type D Reference Data.	
Host database: NPS	
Needs to be running: Runs 7 x 24. Important between 07:00 and 20:00 7 days per week; Critical between 08:00 and 17:30 Monday to Friday and 08:00 and 13:00 Saturday.	
Documentation: [NBXGREV]	

5.2 Application Components for existing Agents

5.2.1 NBS Confirmation Harvester Agent (NB_HV_CONF)

The NBS Confirmation Harvester Agent does not need any enhancement for ICC Transactions to handle the extra payload of ICC-specific data in the [C12] message embedded within the [C1]. The presence of the new <ICCDData:> compound attribute and the extra values in attributes such as Txn_Type do not require any enhancements to the way that the NBS Confirmation Harvester Agent transparently harvests the [C12] message.

There is one exception to this transparency: the maximum size of the data passed to the DRS has to be increased from 2000 to 3000 bytes. The pre-S75 DRS was already configured to handle this increased size, but the interface specification [DRSC12AIS] did not publish this – it has now been increased in the latest version of the specification.

From S75, DRS passes transactions through to TES. It now needs to pass through all transactions, not just financial transactions, and to include the Financial_Transaction indicator with the harvested transactions.

TES also requires the following additional attributes to be harvested. The Retrieval_Reference_Number, one of the primary keys identifying an NBX transaction, is constructed in the same way as is done by the NBX Authorisation Agents, and is done here to localise its construction to the Agents layer. The Logical_FI is the outcome of the same look-up of the Routing_Gateway as the NBX Routing Agent, and identifies which logical NBX Authorisation Agent processed the original transaction. The Entry_Method has to be harvested from within <EPOSSTransaction:> as it does not exist within <Txn:> for banking.

Description	Attribute(s) harvested	Harvested as
Financial_Transaction indicator	<FncITxn:>	<Ctrl.FncITxn>
Date: standard Riposte date of the [C1], in format <i>DD-Mon-YYYY</i>	<Date:>	<Ctrl.Date>
Time: standard Riposte time of the [C1], in format <i>HH24:MI:SS</i>	<Time:>	<Ctrl.Time>
Retrieval_Reference_Number, in format <i>YDDD00nnnnnn</i> , where <i>YDDD</i> is the final digit of the year and the day number within the year, and <i>nnnnnn</i> is the Trans_Num. It is constructed from the Receipt_Transaction_Date and from the penultimate component of the Horizon_Txn_Num (modulo 1,000,000)	<LclDte:> <HTxnNum:>	<Ctrl.RRN>
Logical_FI (e.g. CAPO_A, LINK_B): mapped from the Routing_Gateway using the Routing Data	<RtngGwy:>	<Ctrl.LgclFI>
Entry_Method	<EntryMethod:>	<EtyMde> (under <C12Mess>)

Table 14 – Additional attributes harvested by the NBS Confirmation Harvester Agent

5.3 Unchanged Agents

The following Agents are part of the solution for the NBE Replacement, but require no changes to be made:

- TPS Harvester
- OMDB Heartbeat Harvester

5.3.1 TPS Harvester Agent (T_HV_ALL)

The TPS Harvester Agent was reviewed at S70 and was found not to need any enhancement to handle the extra payload of ICC-specific data in the [C12] message embedded within the [C1].

At S75 the same is true. The presence of extra values in attributes such as Entry_Method do not require any enhancements to the way that the TPS Harvester Agent transparently harvests such attributes.

5.3.2 OMDB Heartbeat Harvester (M_HV_OMDB_HB)

The Low Level Design document for this Agent is [OMDB].

The OMDB Heartbeat Harvester harvests heartbeats written to Riposte by various Agents. The formats of the heartbeats and of the OMDB tables into which they are harvested were Agent-specific for the first two Authorisation Agents, for NBS and DCS. With the ETS Authorisation Agent, new generic formats were devised.

The Heartbeats produced by the new NBX Routing Agent follow this generic format (see [ETSHLD]). The <Application:>, <Data.Heartbeat> and <SysMsgData.Type:> attributes will all be set to "NXRTNG".

The generic Heartbeat and Statistics tables in OMDB are TMS_RX_AUTH_HEARTBEATS and TMS_RX_AUTH_STATISTICS (see [ETSHLD]).

5.4 Interfaces Provided

5.4.1 Interface between NBX Routing and Authorisation Agents

The NBX Routing Agents establish TCP/IP connections to each of the NBX Authorisation Agent instances, both active and standby.

The following message types are supported in the protocol between the NBX Routing Agents and the NBX Authorisation Agents.

Type	Auth direction	Description
RSTS	input	Routing Agent Status. Always the first message on a connection. Used as an 'Echo test request' on a Ping connection. It may be sent at other times, but normally only when the status changes.
ASTS	output	Authorisation Agent Status. Always the first message on a connection, in response to an RSTS. Used as an 'Echo test response' on a Ping connection. It may be sent at other times, but normally only when the status changes.
[R1]	input	Contains timestamp (UTC) when the [R1] was read, together with the [R1] message
[C0]	input	Contains timestamp (UTC) when the [C0] was read, together with the [C0] message
[A3]	output	Contains [A3], together with a flag
[A4]	output	Contains [A4], together with a flag

Table 15: Message types in inter-agent protocol

All messages are delimited by STX and ETX. The first 4 characters indicate the message type. The message is formatted as fields. A field separator character (FS) is used where indicated.

The Routing Agent is responsible for making each socket connection to an Authorisation Agent. It immediately sends a Routing Agent Status (RSTS) message to declare its own status (active or standby) and to elicit the Authorisation Agent's status (active or standby, suspended or not). By comparing information on the Cluster Id and Logical FI the connection is expected to support, the Authorisation Agent can check whether there has been a mismatch in the way the Agents have been configured.

Even a standby Authorisation Agent will respond with an Authorisation Agent Status (ASTS) message. It is the Routing Agent that will decide whether to close the connection or not.

On a connection opened by the Ping thread, the Routing Agent will declare the status as 'Ping'. On a Ping connection, the Authorisation Agent will respond to any RSTS with an ASTS – thus these messages act as an Echo Test Request and Echo Test Response. (Note that there is no need on a Ping connection for the Routing Agent to declare whether it is active or standby.)

Whenever an Agent's status changes, the Agent will immediately report this by sending an RSTS or ASTS message, as appropriate, on all open connections. Note that an Authorisation Agent can send an ASTS at any time without waiting for an RSTS.

Whenever an Agent is about to terminate, it will close all its connections in as tidy a manner as practicable (i.e. using existing techniques).

5.4.1.1 RSTS

A Routing Agent Status (RSTS) message allows a Routing Agent to inform the Authorisation Agent of its status. It is always sent as the first message on a connection. It may be sent at other times, but normally only when the status changes, or as an Echo Test request on a Ping connection. (Note that with the current design the Routing Agent will close a Ping connection once the Echo Test has been performed.)

The Authorisation Agent checks the Cluster Id for being the value expected for this socket. It also checks the Logical FI against its own Identity. In each case a mismatch indicates a configuration error. (These checks need only be made on the first RSTS on a connection.)

No.	Field name	Size	F or V	Value	Description
1	Message Type	4	F	RSTS	Routing Agent Status
2	Active/Standby	1	F	A <i>or</i> S <i>or</i> P	Working connection from an active Routing Agent Working connection from a standby Routing Agent Ping connection
3	Cluster Id	2	F	numeric	Cluster number served by the Routing Agent
4	Logical FI	6	F	left justified, trailing spaces	Logical FI (e.g. CAPO_A)

Table 16: Routing Agent Status (RSTS) message

5.4.1.2 ASTS

An Authorisation Agent Status (ASTS) message allows an Authorisation Agent to inform the Routing Agent of its status (active or standby). It is always sent in response to the first RSTS on a connection. It may be sent at other times, but normally only when the status changes, or as an Echo Test response on a Ping connection.

Note that it is the Routing Agent that is responsible for dropping a connection to the standby if it does not want it, but with the current design the Routing Agent will maintain an open connection with the standby Authorisation Agent.

The active Authorisation Agent uses the Suspended flag to inform the Routing Agent as to whether it may send any (transaction) messages.

- a) A value of "HOLD" tells the Routing Agent to stop sending any such messages and to hold them until the suspension is lifted. With the current failover design, the Authorisation Agent will not in practice issue such a suspension, but the Routing Agent will observe it. Note that a standby Authorisation Agent should set this flag to HOLD, though this is not strictly necessary as a Routing Agent must never send a transaction message to a standby Authorisation Agent.
- b) A value of "ISOK" indicates that a previous suspension, if any, has been lifted. If an Authorisation Agent's status changes in any way, this is immediately reported on all connections.

From its configuration information, the Authorisation Agent knows *a priori* which logical Routing Agent has connected and hence which Cluster this connection is to serve. It includes the Cluster Id in the message, and the Routing Agent checks it for being the one served by the Routing Agent – a mismatch indicates a configuration error.

No.	Field name	Size	F or V	Value	Description
1	Message Type	4	F	ASTS	Authorisation Agent Status
2	Active/Standby	1	F	A or S	Active or standby Authorisation Agent
3	Suspended	4	F	HOLD or ISOK	
4	Cluster Id	2	V	numeric	Cluster number configured in the Authorisation Agent for this socket

Table 17: Authorisation Agent Status (ASTS) message

5.4.1.3 [R1]/[C0]

Used by a Routing Agent to transfer an [R1] or [C0] transactional message to an Authorisation Agent. Such a message may only be sent from an active Routing Agent to an active Authorisation Agent, and then only when the connection is not suspended (Suspended = ISOK).

No.	Field name	Size	F or V	Value	Description
1	Message Type	4	F	[R1] or [C0]	Authorisation or Reversal request
2	Routing Agent Timestamp	14	F	ccyyymmddhhmmss	Timestamp (UTC) of read of [R1]/[C0]
3	FS	1	F	FS	Field Separator
4	Message	2048	V	Riposte message	The [R1]/[C0] message

Table 18: Request messages in inter-agent protocol

5.4.1.4 [A3]/[A4]

Used by an Authorisation Agent to transfer an [A3] or [A4] transaction message to a Routing Agent. It will normally be sent on the same socket connection on which the original [R1] was received. It will always be sent in preference to the active Routing Agent, even if that is on a different connection. But if there is no connection to the active Routing Agent, it will be sent to the standby Routing Agent instead. Note that a standby Routing Agent is capable of receiving [A3]s and [A4]s and writing them to the relevant Riposte group.

The Authorisation Agent may be either active or standby – the latter can occur just after an active Agent has become the standby Agent and there are still [A3] or [A4] messages to flush through to the Routing Agent.

Suspension of the connection does not apply to messages flowing in this direction.

No.	Field name	Size	F or V	Value	Description
1	Message Type	4	F	[A3] or [A4]	Authorisation response
2	Socket Flag	1	F	Space or D	“D” if the [R1] was received on a different socket connection.
3	FS	1	F	FS	Field Separator
4	Message	2048	V	Riposte message	The [A3]/[A4] message

Table 19: Response messages in inter-agent protocol

The Routing Agent will use the Socket Flag to avoid decrementing the count of outstanding responses expected on that connection.

The [A3]/[A4] message does not contain the Riposte red tape as it has not come from a Riposte service. Its format is as follows:

```
<Message:
  <GroupId:group_id>
  <Id:node_id>
  <Application:NBA>
  <Data:
    ...
  >
>
```

5.4.2 Probe Interface to the NBX Authorisation Agents for LINK

Where it is the FI that connects to the NBX Authorisation Agents (only LINK at present), the Agents provide a special port for listening for a connection from a health-check monitor. This monitor 'probes' the Agent to determine whether it could accept an incoming connection from the FI.

The probe is 'successful' if the Agent accepts the connection from the monitor. No data flows across this connection – it is the act of accepting the connection that matters. Therefore, only an active Agent will listen for and accept such connections.

The monitor will disconnect. The Agent will, *in extremis*, disconnect but only when either the connection has been held open too long (10 seconds, configurable by registry) or too many incoming connections have been made (10 connections, configurable by registry).

5.5 Interfaces to External Components

5.5.1 Interfaces to Riposte

5.5.1.1 Riposte Attribute Grammar

[MSGFLOWS] defines the main message flows and message types and acts as a data dictionary for the **Network Banking Application (NBA)**. The reader should also consult the EMV Counter HLD [EMVCTR].

5.5.1.2 NBX Routing Data

From S70, the new **NBXRouting** collection has been defined to hold the Routing Data as to what Transactions are to be handled by NBX. All other Transactions will be processed by the obsolescent NBS Agents and NBE. There is typically one object in this collection for each of NBX's Logical FIs, named after the Logical FI. However, there may be one more than one object for any Logical FI, each with a different name. The `TargetFI` attribute identifies the target Logical FI (e.g. `CAPO_A`) to which the NBX Transactions are to be routed.

The NBX Routing Data is Type D Reference Data. All Agents, both NBS and NBX, read the same data, so minimising any chance of misconfiguration that could occur with registry (which is platform-based). Furthermore, all Agents, both NBS and NBX, access the NBX Routing Data via the same routines, so as to ensure that they take a common view as to which Transactions are to be handled by NBS and which by NBX. Developers should create an **NBX Routing DLL** to achieve this, and to allow for rapid deployment of revised functionality should the need arise.

The Agents read the configuration data only at start-up. The data will normally be non-temporal, but the Agents accept the temporal equivalent. Note that temporal data does not come into effect immediately its start date/time is reached, but only following a restart of the agent.

The Type D reference data will be loaded into the Agent Reference Data dummy offices, 99996*n* for cluster *n*. Although this could allow different data to be loaded into each Cluster, the consistency checks applied to the data require that the same data is loaded into each.

The format is:

```
<Collection:NBXRouting>
<ObjectName: Object_Name>        // e.g. <ObjectName:CAPO_A>
<Data:
  <TargetFI: Logical_FI>        // e.g. <TargetFI:CAPO_A>
  <Filter:
    <RtnGwy: Routing_Gateway>
    <Cluster: value-list>        // Optional: default = all Clusters
    <AgtHash: value-list>        // Optional: default = all Agent_Hash values
  >
>
```

where *value-list* is a comma-separated list of values. The *Filter* attributes are to be logically ANDed. In evaluating the filters involving *value-list*, the filter is TRUE if the value being matched is in *value-list*.

The *Routing_Gateway* value alone is used to decide if the Transaction is to be handled by NBS or NBX. The other filter attributes are used solely by NBX for routing the Transaction to the correct NBX component.

At S75 the Routing Data uses the Agent Hash value alone and not the Cluster number. Mapping Agent Hash values 0 and 2 to the 'A' Authorisation Agents and values 1 and 3 to the 'B' Agents gives 99% accurate load balancing.

5.5.1.3 Heartbeats

The generic format for Heartbeats stored in Riposte and used by the NBX Routing Agent is defined in [ETSHLD] and [OMDB].

5.5.2 Interfaces to FI_EEs

The application and technical interfaces to the external Financial Institutions (FIs) are defined in the relevant AISs and TISs. The mappings between internal Horizon formats for Network Banking transactions and their external counterparts are defined in associated Mapping Documents:

- [CAPO_AIS], [CAPO_TIS] and [CAPO_MAP]
- [LINK_AIS], [LINK_TIS] and [LINK_MAP]
- [A&L_AIS], [A&L_TIS] and [A&L_MAP]

5.5.3 Interfaces to Cryptography and Key Management Services

Cryptography and key management services are used only by the NBX Authorisation Agents. Hence there is no requirement for cryptography support on the NBX Routing Agent Server platforms, which host both the NBX Routing and Guaranteed Reversals Agents.

At S75, NBX introduces new cryptography services to support the management and use of Acquirer Working Keys (AWKs) across the external interfaces to the FI_EEs. These are provided in a new DLL, the **NBX Crypto API** (nbxcryapi) DLL. This DLL uses hardware assistance.

The original cryptography services, such as those to generate and verify digital signatures, continue to be used. These continue to be provided in the **Crypto API** (cryapi) DLL, and still require support for caching.

The high-level descriptions of both NBX Crypto API and the original Crypto API are given in [KMSHLD]. The detailed specifications of the functional interfaces are specified in [CRYPTOAPI] (also in [NBXCRYAPI] and [CRYAPI] respectively). (*Note: The NB Crypto DLL used by the NBS Authorisation Agents is not used by the NBX Authorisation Agents.*)

The performance requirements on both the Crypto DLLs are stringent. It has therefore been necessary for the architectures of the NBX Authorisation Agents and of the Crypto DLLs to be designed to operate harmoniously together. The Agent processing architecture is such that the pools of worker threads are *persistent*, and the DLL's architecture (in particular the NBX Crypto DLL's) is designed to exploit this.

The usage of the crypto functions is now described. Apart from these functions, the Crypto API DLL continues to offer utility functions to convert between Base64-encoding and binary representations of data.

5.5.3.1 NBX Crypto API DLL

The initialisation phase of each Agent process using NBX Crypto API functionality is required to call the **NbxInitialiseCrypto** function. What this actually does is, strictly speaking, outside the scope of this HLD, but it checks that all the resources required by the supported domains are available. However, it does not explicitly wait for the key disk and returns an appropriate response. The Agent retries the function until an OK response is returned (or until total_connection_timeout is reached), outputting suitable NT error events.

The corresponding close function is **NbxTerminateCrypto**.

5.5.3.1.1 AWK management

An NBX crypto session corresponds to one-to-one to a PI Session. The **NbxOpenSession** function is called near the start of a PI Session, before any of the key management functions for handling Acquire Working Keys (AWKs). The corresponding close function is **NbxCloseSession**.

Table 20 lists the usage of the AWK management functions.

NbxGenerateAwk is used to generate the payload for a Key Change message sent by the Agent. **NbxConfirmAwk** completes the transition to a new AWK and is called upon receipt of a Key Change positive response message from the PI.

NbxReceiveAwk carries out the AWK change for a AWK received from the PI on a Key Change message.

NbxKeyTestCheck is used to verify the incoming payload on a Online Key Verification message received from a PI.

Further information on the usage of these functions is in 5.1.3.2.3.

Function	NBX_NBPC_CAPO	NBX_NBPC_LINK	NBX_NBPC_AL
NbxGenerateAwk	Yes	No	No
NbxConfirmAwk	Yes	No	No
NbxReceiveAwk	No	Yes	Yes
NbxKeyTestCheck	No	Yes	No

Table 20: AWK management functions by protection domain

5.5.3.1.2 Translate PIN Block

The **NbxTranslatePinBlock** function is used to verify and decrypt a PIN_Blob_n attribute from the [R1] and re-encrypt it as the PIN_Block for the [R3]. For the [R3], the PIN_Block is protected by the current AWK.

A PIN_Blob is a composite value that not only includes the PIN_Block itself but also a MAC with which the PIN Pad 'signs' the Horizon_Txn_Num. So that this MAC can be verified, the Horizon_Txn_Num is passed in the Transaction_MAC_Data argument to the function.

5.5.3.2 Crypto API DLL

5.5.3.2.1 Verify Digital Signature

The **cryVerifyData** function is used to verify the Digital Signature attribute , <DSig>, on the [R1] and [C0] messages. The associated functions, **cryVerifyStart** and **cryVerifyStop** are also called.

The crypto protection domain is AP.

As the messages will arrive from random Counters, this function will need to cache cryptographic key information in memory to provide the required throughput. The cache needs to be able to hold one key for every Outlet.

5.5.3.2.2 Generate Digital Signature

The **crySignDataEx** function is used to sign the [A3]. The associated functions, **crySignStart** and **crySignStop** are also called. This function contains a parameter to control compression of the Certificate within the Digital Signature.

For both ICC PIN Pad and ICC swipe fallback transactions, this parameter is set to 'force compression'.

For other transactions, this parameter is set to 'compress as determined by [crypto] registry setting'. During migration, the function will return an uncompressed Digital Signature. Once all Counters have been migrated to accept compressed Digital Signatures, the crypto subsystem registry can be configured to get it to always return a compressed Digital Signature.

The crypto protection domain is NBCO.

5.5.3.1.3 Decrypt Data from Outlet

The **cryDecryptData** function is used to decrypt the <Encrypt> attribute in the [R1]. The associated functions, **cryDecryptStart** and **cryDecryptStop** are also called.

The crypto protection domain is NBTDO.

5.5.4 Interfaces to Oracle

The OMDB database is on an NT host. All the others are on Unix hosts.

5.5.5 Interfaces to NPS

The formal definitions of the tables in the NPS are in the NBX Persistent Store HLD [NPS].

The tables used by the NBX Authorisation and Guaranteed Reversals Agents are accessed via synonyms. The tables of interest are:

Table	Table or synonym	Comments
Transaction Status	TMS_RX_TXN_STATUS	Separate table per logical Authorisation Agent
C0 Reversals	TMS_RX_C0_REVERSALS	One table written to by all Guaranteed Reversals Agents and accessed by all Authorisation Agents
C0 Exceptions	TMS_RX_C0_EXCEPTIONS	One table written to by all Guaranteed Reversals Agents
Transaction Journal	TMS_RX_TXN_JOURNAL	Separate table per logical Authorisation Agent
Management Journal	TMS_RX_MGT_JOURNAL	One table written to by all Authorisation Agents
NBX Configuration	TMS_TX_NBX_CONFIGURATION	One read-only table accessed by all Authorisation Agents
NBX Heartbeats	TMS_RX_NBX_HB	Separate table per logical Authorisation Agent
NBX Heartbeats History	TMS_RX_NBX_HB_HISTORY	One table written to by all Authorisation Agents
NBX Statistics	TMS_RX_NBX_STATS	One table written to by all Authorisation Agents
NBX PI Statistics	TMS_RX_NBX_PI_STATS	One table written to by all Authorisation Agents
Agent Checkpoint Table	TMS_ACT_NPS	One table accessed by all Guaranteed Reversals Agents
Agent Run-State Table	TMS_ART_NPS	One table accessed by all Guaranteed Reversals Agents
NPS System Parameters	NPS_SYSTEM_PARAMETERS	One table accessed by all Authorisation Agents
NBX Operator Commands	NPS_OPERATOR_COMMANDS	One table accessed by all Authorisation Agents

Table 21: NPS tables accessed by NBX Agents

5.5.6 Interfaces to DRS

For Network Banking, there is only one direct interface between Agents and DRS:

- From NBS Confirmation Harvester Agent to DRS, for [C12]s

This interface is specified in [DRSC12AIS]. The background to the interface is given in [DRSHLD]. The syntax and semantics of the [C12] message is defined in [MSGFLOWS].

There is also an indirect interface between Agents and DRS. [C1] messages are harvested to the TPS host, from where they are passed into DRS.

5.5.7 Interfaces to OMDB Host

5.5.7.1 Interfaces for OMDB Heartbeat Harvester

<i>These tables are provisionally defined in an Agent LLD [OMDB], awaiting an Interface Specification from the OMDB Host team.</i>
--

5.5.8 Interfaces to TPS Host

At S75 the TPS Harvester continues to harvest [C1] messages for Network Banking into the **TMS_RX_NWB_TRANSACTIONS** table in the TPS database. This table is unchanged, as none of the ICC-specific attributes are harvested.

5.6 Distributed Application Services

The Network Banking Application is a distributed application. At S75 it is described in [DP_NBX].

5.7 Information Management

[MSGFLOWS] defines the main message flows and message types and acts as a data dictionary for the Network Banking Application.

5.8 Networking Services

The network infrastructure is defined in [NBXNETWORK].

5.9 Platforms

The NBX Authorisation Agents send messages to remote FIs. Many of these messages contain an encrypted PIN block, which is hardware-generated. These Agents, therefore, require to be run on NBX Authorisation Agent Server platforms. The Platform Physical Design for the NBX Authorisation Agent Server platform is [PPDAUTH].

The CAPO, LINK and A&L variants of the NBX Authorisation Agents each run on their own platforms. Each of these platforms are of the same general platform type, but each is tailored by means of optional work packages.

The NBX Routing and Guaranteed Reversals Agents share a platform. The Platform Physical Design for the NBX Routing Agent Server platform is [PPDROUTING].

Both the NBX Authorisation Agent Server Routing Agent Server platforms both run Windows 2K.

Other Agents run on Generic Agent Server platforms. Apart from the NBS Confirmation Harvester Agent, they have no special requirements on these platforms. The NBS Confirmation Harvester Agent has to achieve a high throughput, so the platforms have to be specified accordingly.

6. Systems Management

Systems management of the Agents is such an important aspect of the Network Banking Service that it is covered in its own series of documents. Agents also play a role in the systems management of the entire NBX.

6.1 Systems Management of Agents

The active management and recovery of interactive Agent services uses an **Enhanced Agent and Correspondence Server – Resilience and Recovery** (EACRR), see [EACRRSOD].

The main requirement is the ability to replace a failing Agent instance by another one on another platform within a period of the order of 5 minutes (the exact figure is not critical to this HLD) or, for Agent instances which are ‘tied’ by EACRR configuration to a particular platform, within about 1 minute.

6.2 Agents’ Role in Systems Management

The NBX Agents provide many options for being monitored for service management, alerting and other purposes:

- Heartbeats
- Statistics
- NT Events (MONID messages)
- NBX Transaction Journal (in NPS), for individual transactions
- NBX Management Journal (in NPS)

6.2.1 Heartbeats as a Source for Monitoring

6.2.1.1 Heartbeats in Riposte

The harvesting of the NBX Routing Agents’ Heartbeat messages to OMDB provides CS with a rapid source of information on the health of the Routing Agents and, in particular, their interface to the NBX Authorisation Agents. The OMDB Heartbeat Harvester (see 5.3.2) is an important component of this function.

6.2.1.2 Heartbeats in NPS

OMDB is directly monitoring the NBX Authorisation Agents’ Heartbeats in the NPS. This provides CS with a rapid source of information on the health of the Authorisation Agents and, in particular, their interface to the FIs.

OMDB is also gathering information from the Heartbeat History tables in NPS for monitoring and reporting purposes.

6.2.2 Statistics as a Source for Monitoring

6.2.2.1 Statistics in Riposte

The (generic) Heartbeat messages written by the NBX Routing Agents periodically include statistical information.

Each Routing Agent instance maintains statistical information for the items in the following table. The table includes the names of the attributes in the Heartbeat message and of the columns in the OMDB Statistics table TMS_RX_AUTH_STATISTICS.

Description	Attributes in Heartbeat message	Columns in Statistics table
Count of [R1]s and [C0]s received	<R1:<Sum:><New:>>	r1_sum r1_new
Count of [A3]s received from the Authorisation Agents	<EE_A3:<Sum:><New:>>	ee_a3_sum ee_a3_new
Count of messages timed out in the Routing Agent (As there is no response to a [C0], every [C0] will be included in this count.)	<EE_TO:<Sum:><New:>>	ee_to_sum ee_to_new
Count of [A3]s internally generated by the Authorisation Agents, i.e. those with Horizon Response_Code values in the range 30 to 39	<Bad_A3:<Sum:><New:>>	bad_a3_sum bad_a3_new
Total elapsed time (milliseconds) of messages in the Routing Agent, from [R1] received to [A3] written	<AA_ms:>	aa_msecs
Total elapsed time (milliseconds) of messages in the Authorisation Agents and the external FI_EEs	<EE_ms:>	ee_msecs

Table 22: Routing Agent statistics

For the 'count' fields, the Heartbeat contains both the cumulative count since the Agent instance was started and the incremental count since the previous Heartbeat statistics message. For the 'elapsed time' fields, only the incremental value is present. The Heartbeat also records the elapsed time since the previous Heartbeat statistics message.

The Statistics are harvested into OMDB by the OMDB Heartbeat Harvester (see 5.3.2).

6.2.2.2 Statistics in NPS

The NBX Authorisation Agents write statistical information at local midnight and then at a regular thereafter. This will be configured to be every minute. This means that every Agent instance will write its statistics simultaneously, assuming the system clocks are tightly synchronised.

The information is written to two tables in the NPS. One is for statistics gathered for the Logical FI, i.e. for the whole Agent, whereas the other is for statistics gathered per PI.

Each statistic provides a measure for the period just completed (i.e. for a whole or part minute). The Agent does no accumulation over a longer period, such as the lifetime of the Agent instance.

The tables are open-ended in terms of the statistics that may be gathered. The DATA_PERIOD column in each of the two tables may contain a number of statistics values, each of which has the form:

<name:value>

where

- name identifies the statistic, and may be any name that would be valid for an Oracle database column name, with the letters in upper case.
- value is a number, consisting of any number of digits with possibly one decimal point. There may be any number of leading zeros, and if there is a decimal point there may be any number of trailing zeros. Any white space at the start (after the colon and before the

first digit or decimal point) or at the end (after the last digit or decimal point and before the greater-than sign) should be ignored.

In addition, outside these statistics values, the field may contain a small amount of additional comment data that is not intended to be processed or to affect the statistics displayed by OMDB in any way. The only constraint on such additional data is that it will not contain any strings of the form <xxxx>: in which xxxx is the name of a statistic that is to be processed by OMDB. I.e., a search of the whole record for the string <statisticsname: will always find only the statistics value for statisticsname.

The first table below (Table 23) identifies the statistics gathered by the NBX Authorisation Agents identified in [NBXMON] as being required for reports produced by OMDB.

The second table below (Table 24) is for statistics not included in [NBXMON] and that are not expected to be reported by OMDB. They will nevertheless be included in the DATA_PERIOD column of the statistics tables.

Name	Item in [NBXMON]	PI/ FI	Comment	Agent Thread
R3	[R3]s sent	PI		EE_IO
A1	[A1]s received	PI		EE_IO
E1	[E1]s sent	PI		EE_IO
E2	[E2]s received	PI		EE_IO
RJIN	Rejected Transactions	PI	Count of 0620 messages received	EE_IO
DTERR	FI Messages data errors	PI	Count of 0620 messages sent	EE_IO
TOA1	Timed out transactions	PI	Count of [A1] messages that time out	EE_IO
LTA1	Late responses	PI	Count of [A1] messages received when there is no transaction waiting for them. This implies that the [A1] is late (although if the FI made a mistake and sent an [A1] which did not correspond to an [R1] that had ever been sent, it would also be counted here.)	EE_IO
RPTR	Unique repeat reversals	PI	Count of the number of 0421 messages sent for transactions which had not previously had any sent	EE_IO
PINF	Pin Block Failures	PI	Count of [A1] messages whose response code indicates a failure to process the PIN block (currently response code 76)	EE_IO
INFF	FI Transaction Failures	PI	Count of [A1] messages whose response code is one of a set that is classed as indicating an FI infrastructure failure	EE_IO
STAL	Transactions not forwarded to the FI because they were received late by the Agent	FI	Count of stale [R1]s and real-time [C0]s. It includes transactions not forwarded to the FI because they were delayed within the Agent	Verify, PreEE, Exception

LTR	Late Reversals	FI	Count of [C0]s whose transmission times were too late compared with the [R1]s	PreEE
GREV	[E1]s sourced from [C0]s that were received via the Guaranteed Reversals Agent	FI		PreEE
VFR1	[R1] signature verification failures	FI		Verify
VFC0	[C0] signature verification failures	FI		Verify

Table 23: Statistics required for OMDB

Name	PI/ FI	Comment	Agent Thread
R1	FI	Count of [R1]s received from Routing Agent	GetR1
C0	FI	Count of [C0]s received from Routing Agent	GetR1
FIA3	FI	Count of [A3]s generated from [A1]s	GetR1
AGA3	FI	Count of [A3]s generated by the Authorisation Agent itself (e.g. because no [A1] has been received within the timeout, or because of an error in the [R1])	GetR1
AGTM	FI	Sum of the elapsed times in milliseconds from [R1] receipt to sending [A3] for all [A3]s sent by the Agent during the statistics period	GetR1
FITM	FI	Sum of the elapsed times in milliseconds from sending [R3] receipt to [A1] receipt for all [A1]s processed by the Agent during the statistics period	PostEE

Table 24: Other statistics reported to OMDB

6.2.3 NT Events as a Source for Monitoring

NT Events generated by both NBX Routing and Authorisation Agents are monitored to provide information on the health of the service itself and of the resources upon which the Agents are dependent. This monitoring system is described in [NBMON].

The events monitored are all Error events from the Agents. In addition there are a class of events specifically generated for this purpose. Each resource being monitored is identified by its “**monitor id**”, introduced in the text of the event message by the keyword “MONID:”. The monitor id is limited to 20 characters. The resources being monitored, and their associated monitor ids, are given in the following table, where *short_service_name* is the NT service name omitting the “TMS” prefix.

Note that resource CS1 refers to the preferred Correspondence Server, CS2 to the non-preferred one. Only one of CS1 or CS2 needs to be available for an NBX Routing service to be able to run successfully. Note similarly that resource DB1 refers to the preferred Oracle instance, DB2 to the non-preferred instance. Only one of DB1 or DB2 needs to be available for an NBX Authorisation service to be able to run successfully.

Resource	Monitor id
The NBX Routing service itself	<i>short_service_name.SERVICE</i>
The NBX Authorisation service	<i>short_service_name.NBX</i>
Riposte: Correspondence Server <i>n</i>	<i>short_service_name.CSn (n = 1, 2)</i>

Table 25 – Monitor Ids of the Resources for the NBX Routing Agent

Resource	Monitor id
The NBX Authorisation service itself	<i>short_service_name.SERVICE</i>
PI xxxx in the 'Enquiry Engine' [The PI's name, xxxx, in practice starts with "PI"]	<i>short_service_name.xxxx</i>
Crypto	<i>short_service_name.CRYPTO</i>
NPS instance <i>n</i>	<i>short_service_name.DBn (n = 1, 2)</i>
System clock	<i>short_service_name.CLOCK</i>

Table 26 – Monitor Ids of the Resources for the NBX Authorisation Agent

Each event has an associated "severity level", for which the keyword is "MONSEV:". The severity levels that will be used are given in the following table.

Severity Level	Value
Good	G
Warning	W
Bad	B
Information	I

Table 27 – Monitor Severity Levels

Once a resource has been flagged as Bad or Warning, it is necessary to specifically flag it as Good to clear its monitoring system's view of the resource's status. During Agent start-up, all resources will be flagged as Good in order to clear any possible previous Bad or Warning settings.

The following table gives the meanings of the monitored NT events. To aid the filtering process, they will be assigned their own facility code and will use a reserved range of event numbers (8000 to 9999).

Resource	Severity level	Description
The NBX Routing service itself	Good	Service is available.
	Bad	Service is closing: <i>reason</i> .
	Information	Service is Active.
	Information	Service is Standby.
The NBX Authorisation service	Good	NBX Authorisation Agents are available.

	Bad	Some or all NBX Authorisation Agents are unavailable: <i>reason</i> .
Riposte: Correspondence Server <i>n</i>	Good	<i>CSn_host_name</i> is available.
	Bad	<i>CSn_host_name</i> is unavailable: <i>reason</i> .

Table 28 – NT Events for Monitoring the NBX Routing Agents

Resource	Severity level	Description
The NBX Authorisation service itself	Good	Service is available.
	Bad	Service is closing: <i>reason</i> .
	Information	Service is Active.
	Information	Service is Standby.
Target <i>Plxx</i> (e.g. <i>PIA1</i> , in upper case)	Good	<i>Plxx</i> is available.
	Bad	<i>Plxx</i> is unavailable: <i>reason</i> .
Crypto	Good	Crypto facilities are available.
	Bad	Crypto facilities are unavailable: <i>reason</i> .
NPS instance <i>n</i>	Good	<i>NPS_instance_n</i> is available.
	Bad	<i>NPS_instance_n</i> is unavailable: <i>reason</i> .
System clock	Good	Clock drift from NPS's clock is OK.
	Bad	Clock drift is above threshold.

Table 29 – NT Events for Monitoring the NBX Authorisation Agents

6.1.4 NBX Transaction and Management Journals

The NBX Authorisation Agents record the following to both the Journals for systems management purposes:

- All Network Management (0800 & 0810) messages sent and received (see 5.1.3.2.3)
- PI availability records (see 5.1.3.2.4)
- TCP/IP Connection status records (see 5.1.3.2.2.5)
- All Administration Advice (0620) messages sent and received, together with the bad message being rejected when the Agent generates an 0620

In addition, the Agents record the following in the Management Journal only:

- All operator commands (see 5.1.3.3)

7. Application Development

The normal Agent development environment is adequate for most of the development and unit testing.

A 'stub' version of the NBCryptoAPI DLL is required from IPDU for the unit testing of the NBS Authorisation and NBS Expedited Confirmation Harvester Agents in the absence of hardware crypto cards.

An emulator for CAPO, LINK and A&L is provided by Lexcel. This will provide a suitable environment to unit test the NBX Authorisation Agents.

7.1.1 Testing options for volume and performance testing

The agent supports some testing options for volume and performance testing, but these are only available on a test environment. They will not be supported in the Live (or LST) environment. In particular these options allow:

- signature verification failures to be ignored, or signature verification to be bypassed;
- an override of the HTxnNum to be used when the PIN blocks are verified/translated in the [R1].

These options are only available through the AgtNBAuthTestMode.dll. The AgtNBAuthTestMode.dll is a test tool delivered through PVCS for installation on test rigs only. It is used only by this Authorisation agent, and is activated through a registry value 'TESTMODEDLL'.

There are no required changes to this DLL. *(The requirement to increase the number of override entries from 2000 to 5000 has evaporated.)*

8. System Qualities

8.1 Resilience

8.1.1 Resilience to a failing Correspondence Server

The NBX Routing Agent is required to continue functioning when the Riposte connection to the Correspondence Server fails, either because the Riposte service itself fails or because the LAN connection to it fails. The Agent is required to dynamically switch to accessing the Riposte service on an alternative Correspondence Server. The requirement and solution are the same as for the NBS, DCS and ETS Authorisation Agents. The NBX Routing Agent will be configured with a Resilient Locale.

The NBX Guaranteed Reversals Agent will also be configured with a Resilient Locale. If the Riposte connection to a Correspondence Server fails, this Agent will fail and wait to be restarted by Tivoli/EACRR mechanisms. During the connection phase, it will attempt to connect to the preferred Correspondence Server, but if that is not possible it will connect to the alternative instead. Note that this requirement and solution is the same as for the NBS Expedited Confirmation Agent that it supersedes.

8.1.2 NBX Routing Agent

See 5.1.2.6.

8.1.3 NBX Authorisation Agents

See 5.1.3.8.

8.1.4 NBX Guaranteed Reversals Agent

See 5.1.4.6.

8.2 Performance and Scalability

8.2.1 NBX Routing Agent

See 5.1.2.5.

8.2.2 NBX Authorisation Agents

See 5.1.3.7.

8.2.3 NBX Guaranteed Reversals Agent

See 5.1.4.5.

8.3 Security

8.3.1 Service Users

From BI3 the Secure Build Implementation Guide, [SECBUILD], imposed new standards on Service Users. It is no longer permissible for platform owners to define Service User accounts that default to use the administrator level of privileges associated with the local system account (LSA). Instead, all Service User accounts must be created using the global group and local group model with the minimum level of privileges being assigned in order to

achieve the required functionality. The standard for such Service Users is to create them in the local resource domain as domain users.

The standards apply to the new NBX-specific Agents which run as NT services. Table 30 lists the Service User names – these are copied from [SECROLES], which is the definitive document. Note that the NBS Confirmation Harvester Agent was first introduced at Release S30, and the information pertaining to it is unchanged.

Agent	Platform	Service Name	Service User
NBX Authorisation Agent for CAPO	NBX Agent Server	TMSNX_CAPO_sfx	TMSNXAuth
NBX Authorisation Agent for LINK	NBX Agent Server	TMSNX_LINK_sfx	TMSNXAuth
NBX Authorisation Agent for A&L	NBX Agent Server	TMSNX_AL_sfx	TMSNXAuth
NBX Routing Agent	NBX Routing Agent Server	TMSNXRtnsfx	TMSNXRouting
NBX Guaranteed Reversals Agent	NBX Routing Agent Server	TMSNXGRevsfx	TMSNXGRev
NBS Confirmation Harvester Agent	Generic Agent Server	TMSNBConfSfx	TMSNBConf

Table 30 – Service Users for NBX-specific Services

8.4 Potential for Change

The design has attempted to build in the potential for change so far as is practicable. For example, it has continued the approach adopted with previous Agents of making as much as possible configurable.

As stated in the Design Principles, design decisions have taken account of the requirements concerning the potential for change. One of the requirements is that much of the configuration of the NBX Authorisation Agents must be “soft”. This particularly applies to timers, such as the time of the end of the business day, and to protocol mappings, such as the mapping from FI response codes to Horizon response codes.

See 5.1.1.4 for an overview of what is configurable and by what mechanism.

The following specific items have potential for change:

- NBX Authorisation Agent could apply back pressure to NBX Routing Agent. The inter-Agent protocol allows for it and the Routing Agent behaves correctly.

9. Solution Implementation Strategy

All functionality described is to be released at S75 except where specifically mentioned to the contrary.

10. Migration

10.1 Migration from NBE to NBX

The migration strategy for migrating network banking transactions from NBE to NBX, its replacement, is covered in the S70/75 Migration Strategy, [MIGRATION]. Most of it need not be rehearsed here again.

As described in earlier sections, NBX Routing Data is used to distinguish between messages to be routed to NBE and those to be handled by NBX. During the period of co-existence, NBX Agents will silently ignore messages that are not explicitly to be handled by NBX. Once migration is complete, a change to a registry setting will cause messages with an unexpected Routing_Gateway value to be errored in the event log.

10.2 Migration of Individual Agents

There are believed to be no migration issues with individual Agents.