Document Title:	**Generalised API for OPS/TMS: Appendix C System Management**
Document Type:	Technical Design Standard
Release:	N/A

Abstract:

This appendix describes System Management interfaces to be used, and functions that must be supported by applications developed for the Horizon environment.

The main document provides the information required to plan the development of new applications and describes in more detail the architecture set out in the OPS Architecture Specification.

Both documents are supplied under the terms of the Codified Agreement to POCL to facilitate the procurement of applications to run on the Service Infrastructure (interfacing with OPS and TMS).

This document is only available to organisations outside ICL Pathway through formal Non-Disclosure Agreement.

Document Status:	APPROVED
Author & Dept:	Jon Cruise, Technical Design Authority
Contributors:	Tony Hayward, Janet Dore
Reviewed By:	ICL Pathway: Terry Austin, John Dicks, John Allen, Simon Fawkes, Allan Hodgkinson, Glenn Stephens, James Stinchcombe, Dave Tanner, Peter Wiles POCL: Bob Booth
Comments By:	N/A
Comments To:	Document Controller & Authors
Distribution:	ICL Pathway Library and Reviewers

0 Document control

0.1 Document history

Version No.	Date	Reason for Issue	Associated CP/PinICL No.
0.10		First version of this appendix.	
0.11	7/2/2000	Second version	
0.12	24/2/2000	Comments from earlier versions incorporated.	
0.13	29/2/2000	Comments from earlier versions incorporated.	
0.14	21/3/2000	Comments from earlier versions incorporated.	
0.15	30/3/2000	Comments from earlier versions incorporated.	
0.16	3/4/2000	Comments from earlier versions incorporated.	
0.17	17/4/2000	Comments from earlier versions incorporated.	
0.18	3//5/2000	Comments from earlier version incorporated.	
1.0	5/5/2000	Issued for approval.	

0.2 Approval authorities

Name	Position	Signature	Date
T. Austin	Development Director		
J. Dicks	Customer Requirements Director		
R. Booth	POCL		

0.3 Associated documents

Reference	Version	Date	Title	Source
TD/ARC/029	0.4	12/11/99	TMS Architecture Specification	ICL Pathway
TD/ARC/030	0.4	12/11/99	OPS Architecture Specification	ICL Pathway

0.4 Abbreviations and definitions

Abbreviation	Definition
AIS	Automated Interface Specification
BST	British Summer Time
CM	Configuration Management
CP	Change Proposal, in ICL Pathway change management process
CSV	Comma Separated Variable
DLL	Dynamic Link Library
EPOSS	Electronic Point of Sale System
OBC	Operational Business Change
OLA	Operating Level Agreement
OPS	Office Platform Service
PinICL	Record of a problem in ICL's Problem tracking system
POCL	Post Office Counters Limited
PVCS	Configuration management tool from Intersolve Inc.
RIPOSTE	Retail Integrated Point Of Sale system in a Transaction Environment: product from Escher that provides both the infrastructure and the Desktop environment of the Horizon system. The definitions in this manual refer to version 6 onwards
SLA	Service Level Agreement
WAN	Wide Area Network

0.5 Changes in this version

Version	Changes

0.11	Review comments from version 0.10 have been incorporated.
0.12	Review comments from version 0.11 have been incorporated.
0.13	Review comments from version 0.12 have been incorporated.
0.14	Review comments from version 0.13 have been incorporated.
0.15	Review comments from version 0.14 have been incorporated.
0.16	Review comments from version 0.15 have been incorporated.
0.17	Review comments from version 0.16 have been incorporated.
0.18	Review comments from version 0.17 have been incorporated.
1.0	Review comments from version 0.18 have been incorporated. Issue for sign-off.

0.6 Changes expected

Changes
1. This document reflects the current implementation. The provided descriptions and definitions may be subject to change control as determined by technical and/or operational needs.

0.7 Table of Contents

C.1 Introduction

The Horizon system developed by ICL Pathway is a secure environment incorporating a number of application services. It is subject to Service Level Agreements (SLAs) governing the performance and response of each of the services involved.

The development and implementation of a new application must not cause any adverse effect on the existing service, or on the applications within each service, thus compromising these Service Level Agreements. Neither must it jeopardise the security of the Horizon environment.

In addition, a number of Operating Level Agreements (OLAs) have been established. These support SLAs by detailing operational timetables and support arrangements.

Despite defining the APIs needed to develop new applications in this environment (see main document), system management considerations dictate that there may be an impact on the management of the overall system. The system management characteristics of the new application must therefore be established, as must its performance characteristics.

The scope of the system management domain is described in section C.2. The scope covers the components and documentation that must be provided to support the system management life cycle. This life cycle includes the integration, testing, release, and maintenance of new applications.

C.2 Scope

The existence of the Horizon systems management framework imposes certain constraints on the behaviour of applications and the way they are constructed. The phases system management life cycle are shown in Figure C-1. For new applications the key elements are:

- *Application Configuration* where the application first enters the system management domain. At this point all initial discussions and reviews must have been completed and a definition of the characteristics of the application established.
- *System Integration* where the application is integrated into the rest of the Horizon system. At this point in the life cycle the key constraints are the way the software is packaged, its dependencies on other components, any security implications, its relationship to existing Reference Data, and its need for new Reference Data.
- *System Test* where the application is tested in an end to end system with the same configuration as in the live estate. Again, software packaging is an important element as this is the first point at which the live configuration is tested. Security is also important as it may have operational implications. The way the system is used in the outlet, as well as the support capability of the application and its performance against predictions, are also tested at this point.
- *Application Installed and Implemented* covers the installation of the application throughout the live estate and the initiation of its use by staff at each outlet. The co-ordination of the installation of the software and Reference Data components and the user procedures and the training schedule, are the key elements at this point together with the ability to regress the implementation should errors occur.
- *System Supported* where the support of the application, its performance in the live estate, and the way any error fixes are introduced into the estate become significant, as does the training the users receive.

The way the various system management components fit into the overall life cycle is described in section C.2.1. There are implications for the roles and responsibilities of Third Parties, POCL, and ICL Pathway. These are identified in section C.2.2 with the way they map onto the life cycle covered in section C.2.3.

Irrespective of which organisation is responsible, a series of reviews must be conducted to allow the resolution of any issues identified to be embodied in the new applications, thus ensuring that the best solution can be implemented to the benefit of the whole system. These reviews are described in section C.2.4.

One of the main elements of system management is the distribution process itself. Section C.3 covers the distribution of software and Reference Data. The mechanisms used in each element of the life cycle, and the documentation needed to support them, are covered in the remaining sections. The implications of user support, training and procedures are included where appropriate.

C.2.1 System management environment

The diagram below, Figure C-1, defines the components that must be provided to support each phase of the System Management Life Cycle.

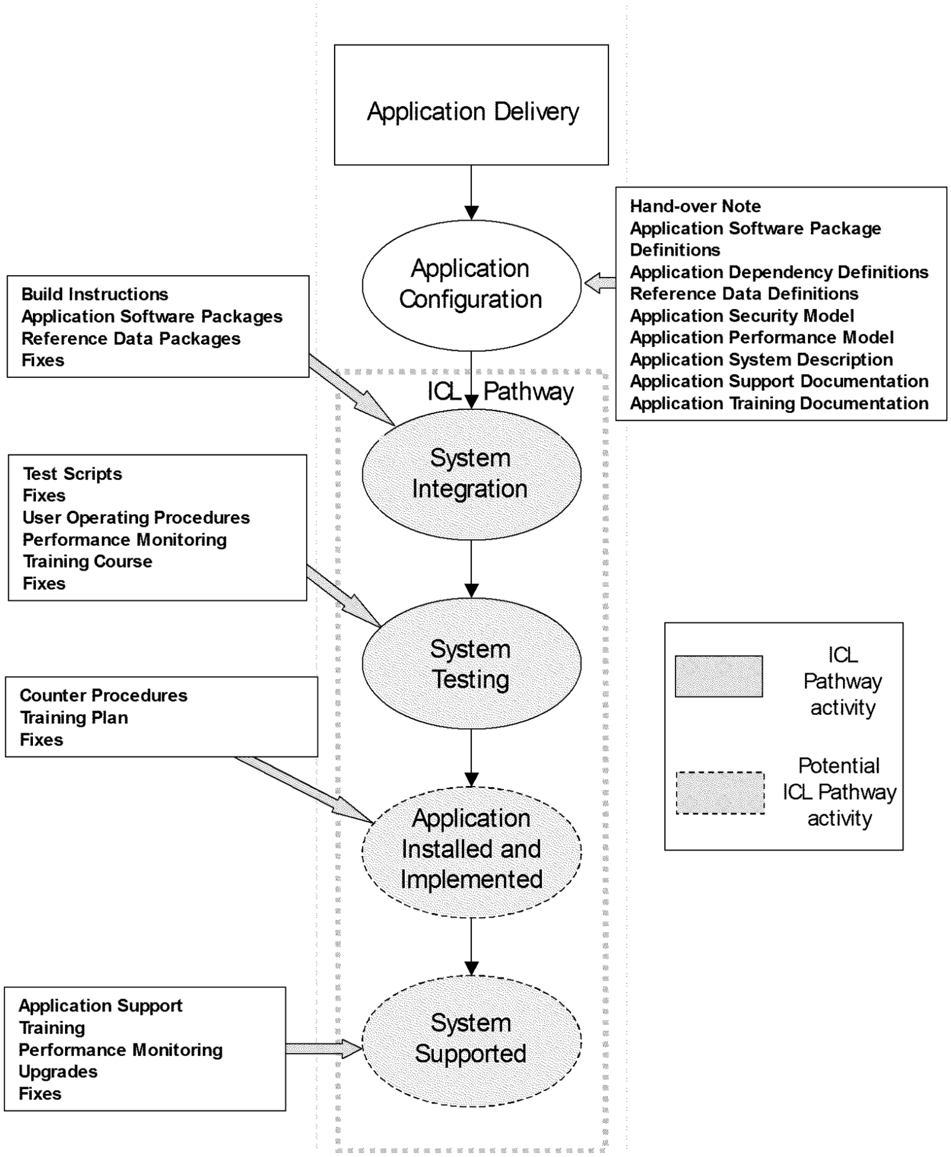
Application Delivery
Application Configuration
Hand-over Note
Application Software Package Definitions
Application Dependency Definitions
Reference Data Definitions
Application Security Model
Application Performance Model
Application System Description
Application Support Documentation
Application Training Documentation
Build Instructions
Application Software Packages
Reference Data Packages
Fixes
ICL Pathway
System Integration
Test Scripts
Fixes
User Operating Procedures
Performance Monitoring
Training Course
Fixes
System Testing
ICL Pathway activity
Potential ICL Pathway activity
Counter Procedures
Training Plan
Fixes
Application Installed and Implemented
Application Support
Training
Performance Monitoring
Upgrades
Fixes
System Supported

Figure C-1 System management environment

Figure C-1 identifies those activities within the system management life cycle that ICL Pathway will undertake and those that ICL Pathway may be subcontracted to undertake by a third party, see C.2.3.

If the third party supplier does not subcontract ICL Pathway to install and implement its application, then the *Risk Review*, will be held to determine the interactions with the ICL Pathway integrated systems.

C.2.2 Roles and responsibilities

The roles and responsibilities of those involved in the supply, integration, testing, implementation and support of any new application must be agreed and the implications reviewed at the start of and during the implementation of a new application.

The roles and responsibilities currently agreed with POCL are given in section C.2.3 where the implications for system management are identified. To support these roles and responsibilities, a number of reviews are needed. The objectives for each review and the main topics addressed are given in Table C-2, in section C.2.4.

ICL Pathway will assist POCL in identifying the roles and responsibilities of all parties involved with third party software. The areas where agreement is required include:

- Licensing arrangements
- Testing policy
- IPR
- Ongoing support and modification
- Management of Change
- Liabilities, Cancellation and Termination
- Fault Management
- Service Introduction

C.2.3 Life cycle responsibilities

The table below sets out ICL Pathway's view of its responsibilities through the life cycle of a new third party application being introduced on to OPS/TMS.

Activity	***ICL Pathway responsibility/role***
Programme Management	ICL Pathway will not be responsible. ICL Pathway will provide suitable representation at Programme meetings. This activity will contain the definition of additional documentation or services to be provided by ICL Pathway.
Business Requirements	ICL Pathway will not be responsible, but wishes to be involved from the earliest opportunity.
Systems Design	ICL Pathway will be responsible for the overall design.
Application Design	ICL Pathway will not be responsible.
Application Development	ICL Pathway will not be responsible.
Application Test and Integration	ICL Pathway will not be responsible. *This is the unit and link testing of the application in isolation from other applications.*

Systems Integration	ICL Pathway will be responsible.
Systems Test	ICL Pathway will be responsible. *This is the end to end testing of the system with the new application integrated into the same configuration as the live estate.*
Type Approval	ICL Pathway will be responsible for hardware and for software type approval.
Business Acceptance	ICL Pathway will not be responsible.
Manufacture	For hardware ICL Pathway may or may not be subcontracted by the supplier.
Distribution	For hardware ICL Pathway may or may not be subcontracted by the supplier.
Installation and Implementation	For hardware ICL Pathway may or may not be subcontracted by the supplier. For software ICL Pathway may or may not be subcontracted by the Supplier.
Support	ICL Pathway may or may not be subcontracted by the Supplier. All software changes must be integrated and fully validated by ICL Pathway. *(Support means software fault detection and rectification.)*
Maintenance	For hardware ICL Pathway may or may not be subcontracted by the third party supplier. *(Maintenance means hardware maintenance.)*
Training	ICL Pathway may or may not be subcontracted by the Supplier.
Help Desk support	ICL Pathway may or may not be subcontracted by the Supplier.
Service Reporting	ICL Pathway may be responsible for reporting but will not take Service Level Agreement responsibility for the application. *This means ICL Pathway may be responsible for producing reports of performance against SLAs.*
Invoicing	ICL Pathway may provide information on which an invoice can be based, but will not be responsible for invoicing or reconciliation. *This means ICL Pathway may provide information in the form of reports from which invoices may be prepared.*

Table C-1 Activities and responsibilities

C.2.4 Reviews

System Management starts after the software and data that comprise the application are delivered as packaged components. Before any application can be introduced into the system management environment, a number of reviews must have been conducted:

- *Dependencies Review.*
- *Performance Review.*
- *Process Review*
- *Risk Review.*
- *Security Review* (see also Appendix B, Cryptography and Key Management).
- *SmartMan Review (for smart card applications)* (see also Appendix A, SmartMan Interfaces).
- *System Management Review.*

Note that these reviews may need to be repeated after the application has been delivered.

Review	***Objective and Main Topics***
Dependency	Establish the dependencies of the new application on the other parts of the Horizon system. The main topics are: • Synchronisations. • Component lists. • Dependencies.
Performance	Define the expected performance of the system so that the Performance Model can be established and to support its validation during System Test. The main topics are: • Rate of change of software. • Rate of change of Reference Data. • Use of the General Acknowledgement Agent. • Expiry periods. • Volumes of transactions, Reference Data changes and resource usage. • Frequency of events. • Use of persistent objects. • Use of indices.
Process	Establish and document the processes carried out by the clerk at the counter. The main topics are: • Counter Procedures. • Help provided to the counter clerk.
Risk	Establish and quantify the risks where ICL Pathway is not being contracted to install and implement a new application. The main topics are: • Interfaces. • Interactions. • Impacts.
Security	Establish and quantify the impact of existing security facilities, and the scope of any changes required. The main topics are: • Access control. • Domains. • Security log.

SmartMan	Identify and quantify the impact of introducing a new smart card application, including: • Smart card technology to be used, whether it can be supported. • Firmware enhancement (of the card reader/encoder). • Impact on existing SmartMan implementation. • Recovery conditions. • Use of menus.
System Management	Identify all components and the way they will be integrated, tested and distributed, together with the support needed during System Integration, System Testing and live operation. The main topics are: • Installation, implementation and support. • Restarts, and manual intervention. • Software distribution. • Audit Service. • Additional support from the Horizon Help Desk. • Separate Help Desk for the new application.

Table C-2 Review objectives

The result of these reviews are agreed deliverables, interfaces, and documentation that provides the level of understanding that allows the application to be configured and delivered into the System Integration testing. After the integrated system has been System Tested, the application can be released into the live environment and become part of the supported system.

C.3 System Management and Package Distribution

Software has to be packaged so that it can be distributed effectively in the distributed environment of 20,000 Post Office outlets.

The primary software distribution tool used in the Horizon solution is Tivoli. Application Products and Items and their design documentation must be registered in PVCS. The registered application Product and Items can then be converted into Tivoli file packages and installed (and if necessary un-installed) by Tivoli in unattended mode. The overall software distribution process is described in section C.3.1, with the way each package is installed covered in section C.3.2. The way the installation is monitoring is also described in section C.3.2, as is the way the inventory is maintained.

Some Reference Data is distributed through the Reference Data Management Centre and the rest as work packages. These distinctions are described in section C.3.3.

There are also rules that have to be followed when creating packages for distribution. These are covered in section C.3.4.

C.3.1 Overall software distribution processes

The same packaging is used during integration, testing, and the maintenance of the live estate. The scope of the Software Distribution process is as follows.

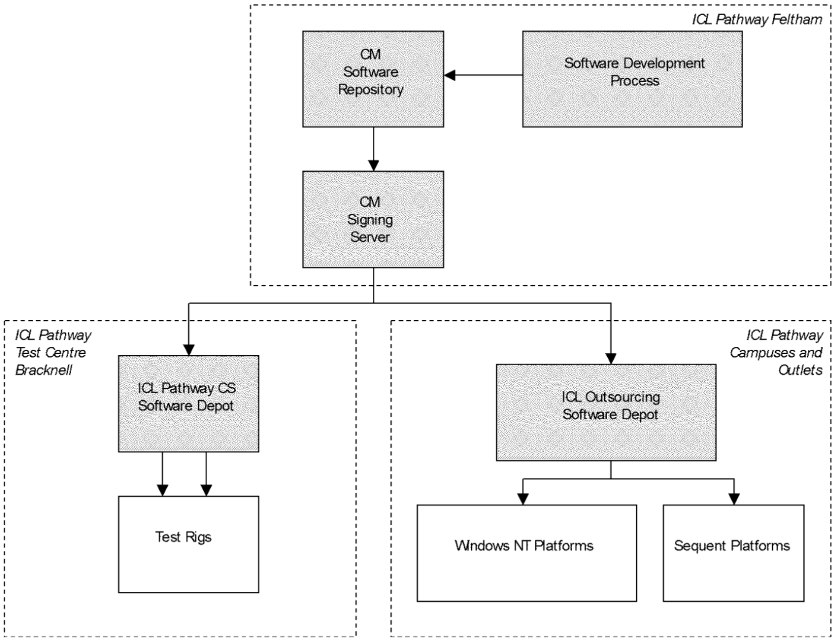
ICL Pathway Feltham
CM Software Repository
Software Development Process
CM Signing Server
ICL Pathway Test Centre Bracknell
ICL Pathway CS Software Depot
Test Rigs
ICL Pathway Campuses and Outlets
ICL Outsourcing Software Depot
Windows NT Platforms
Sequent Platforms

Figure C-2 - Software Distribution

Software is lodged in the ICL Pathway Configuration Management (CM) Software Repository. It must have an identifiable Name and Version Number, see section C.4.1 and section 8 in the main document. These are carried with it throughout the distribution process.

When it is ready for release, it is transmitted via the CM Signing Server to either the ICL Pathway Software Depot for testing at Bracknell, or the Outsourcing Software Depot for distribution to the live estate. In both cases, it is first verified as being ready for release. The purpose of the distribution to Bracknell is to verify that the Tivoli distribution packages can be applied successfully, and that the package can be regressed if necessary.

The CM Signing Server ensures that all packages are signed so that any corruption can be detected on arrival at the outlet. The base Riposte software is separately signed to ensure that only the correct OPS environment is loaded at each counter position, see section C.3.2.3.

C.3.2 Software distribution and inventory management

C.3.2.1 Software distribution monitoring

Tivoli *Distribution Monitoring* runs on each client platform and monitors the operation of the installation process. It returns status information to the inventory, which records the system status and the presence of the installed files.

C.3.2.2 Software inventory management

The inventory uses a central Oracle database within the systems management function to maintain details of the platforms, software and applications asset base in the installed OPS. It includes a Web browsing capability that enables support staff in a wide variety of locations to examine the status of distributions, the software state of any managed node, and any problems that have occurred with a distribution. The database is also used by other system components, such as the software distribution service.

The database is hierarchically structured and is extensible to cope with any new hardware or software objects or attributes deployed in response to future services or applications needs. It is updated automatically when any new equipment is installed in an outlet, or when new or updated software is distributed and installed.

Any changes needed to support the introduction of a new application must be raised at the *System Management Review*.

C.3.2.2.1 Inventory signatures

For security and integrity purposes, signatures are applied to platforms and products:

- *Platforms* Each counter has a signature identifying the base build software. This signature is also held in the inventory for that counter on the central management system.
- *Products* Each change to a product on a particular counter has an unique signature held on the counter itself and also in the inventory for that counter on the central management system.

These signatures are maintained by the ICL Pathway distribution mechanism. The signatures name one or more constituent files. All of these signatures on a counter can be remotely inspected from the ICL Pathway central management system.

C.3.2.3 Signing on distribution

All application software packages are signed for distribution to outlets. Software that is distributed is signed by the CM Signing Server (see C.3.1). Additionally, those EXEs, DLLs, and files associated with the Riposte Desktop and message server and any security related software are individually signed and the signatures are checked each time the code or file is loaded. This ensures that the correct environment for applications is established each time that the Desktop is loaded. This signing is in addition to the signing of work packages for Tivoli distribution purposes.

C.3.3 Reference Data distribution

Reference Data is used throughout the Horizon system for a variety of purposes. It is used to support the business processes in the outlets in the live estate, client operations as well as determining the products and pricing that support the generation of Cash Account reports at each outlet. It also defines the interpretation of the use of counter peripheral, as well as defining the dialogue that controls the actions carried out by clerks for individual products and services.

Reference Data is fundamental to the accounting integrity of the system. The make up of the cash account is defined through Reference Data, for instance. Reference Data also plays a fundamental role in roll out, for example during migration.

C.3.3.1 Reference Data interfaces

For each of the five types of Reference Data used by an application a separate *Reference Data* interface is used, see section 3.1.2, *Data interface types*, in the main document.

C.3.3.1.1 Business Type A

Type A is POCL Reference Data that conforms to the POCL Automated Interface Specification (AIS).

C.3.3.1.2 Business Type B

Type B data originates from POCL but is not included in the automatic AIS interface. This data is transferred in different formats (e.g. spreadsheets, CSV files, text documents as part of the Operational Business Change process).

C.3.3.1.3 Implementation Type C

This is implementation data generated entirely by ICL Pathway, often at the behest of POCL.

C.3.3.1.4 Implementation Type D

This data is also known as ‘Build Objects’ in that it is Reference Data generated by ICL Pathway concerning the underlying Horizon build. It contains Riposte Objects and Training data (to enable Training Mode to be entered from the Live Desktop).

C.3.3.1.5 Client Reference Data

This interface is implemented by the application concerned. An example of this kind of data is tariff data.

C.3.3.2 Distribution of Reference Data

C.3.3.2.1 Delivery of Business Reference Data

The ICL Pathway Reference Data Management Centre (RDMC) holds POCL supplied Business, and some Implementation, Reference Data in an Oracle database. This database acts as the source for the distribution process. The Reference Data Distribution Server (RDDS) creates agent views of the Reference Data by pre-processing the Reference Data into attribute grammar form. Such Reference Data is held in (attribute grammar) ‘Collections’, each Collection containing a set of related Objects (themselves decomposing into Attributes). Figure C-3 illustrates the distribution mechanism.

Business Reference Data is held as *temporal persistent* objects. Temporal objects have a start date and (optional) end date. Changes to temporal objects are recorded by end-dating one instance of that object, and adding a new occurrence with new start and end dates.

Where any item within an object has changed, the pre-processing stage creates a new version containing all the items within that object that apply from the date that the change is valid (allocating this a new version number).

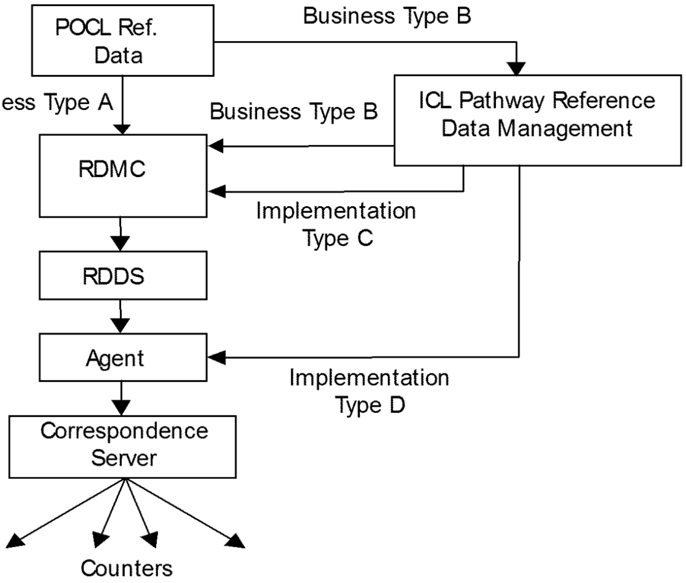

Figure C-3 Reference Data distribution

C.3.3.2.2 Delivery of Implementation Reference Data

Implementation Reference Data, that is Type C and Type D data, is held as persistent objects in collections, but the data is not temporal in that it remains extant until either replaced or deleted.

Reference Data used by new applications will be provided as a set of collections using the naming conventions and collection names identified in sections 8 and 9 of the main document. The only exception to this rule is the Reference Data associated with Access Control, (see section C.5).

The arrival of specific Reference Data at an outlet can, if necessary, be monitored by use of the General Acknowledgement Agent, see section 7.2.1, *General Acknowledgement Agent*, of the main document for details.

Because of the performance implications of such an approach, any usage would need to be quantified during the *Performance Review*.

C.3.3.3 POCL Reference Data change authorisation procedures

The complete list of all Reference Data changes allowed for Types A, B, and C is set out in the *Reference Data Change Catalogue* (CS/IFS/001). This categorises changes into three types: basic (planned), advanced (planned) and unplanned.

C.3.3.3.1 Basic Changes

Basic Changes are those that ICL Pathway can support without prior notice and only require basic control and release activity by ICL Pathway. The data contained in these is referred to as 'Class1' data, for example a change to an outlet's telephone number or change in product price.

All Basic Changes are supplied by POCL as Type A data and are released without reference to the Operational Business Change (OBC) process that controls releases of changes into the live estate.

C.3.3.3.2 Advanced Changes

Advanced Changes are those that need additional activity by ICL Pathway before the change can be implemented.

Advanced Changes are always authorised via the OBC process, and are supplied via Type A, B, or C Reference Data, or could be implemented via Type D Reference Data.

C.3.3.3.3 Product Changes

A Product Change is the change needed to the definition of a product. It is always supported through Reference Data and may be either a Basic or an Advanced Change. Table C-3 identifies the relationships between categories of change and types of Reference Data involved. Scales Reference Data is supplied via Type B Reference Data, and is considered as product data.

Categories	*Type*	*Examples*
Basic Change - Pure	Type A	Change clerk instructions (and possibly changing non-core product availability).
Basic Change – High Risk	Type A	Product price, product names, revaluation, method of payment, change whether voidable/reversible, change min/max quantities/values.
Advanced Simple Change	Type A plus after the event changes	Cease non-value product, Non core product becomes core, Change to use of additional fields.
Advanced Standard Change	Type A and Type B (also MIS change)	Change to ability to print receipt, change value to non-value stock, picklist, remove AP Client, calendar.
Advanced Complex Change	Type C	New product, change scales tariff, screen layout, change accounting node, change product core to non-core.
AP Changes	AP Client Take	New AP Client, client name change,

	On	cease AP Client, product or token.
Fastrack – Basic Express	Type A	Price of non-value or re-valuable value stock, change product name.
Fastrack – Migration Special	Type B	Correcting Reference Data to show non-core products at outlets.
Fastrack – Tight Timescales	Any	Price changes.
Fastrack – Error Correction	Any	Any.

Table C-3 Product changes

C.3.3.4 ICL Pathway Reference Data change authorisation procedure

For Types C and D Reference Data, changed or new data is released via work packages and follow the normal OBC procedure for authorisation of the work package into the live estate.

C.3.4 Software and Reference Data packaging rules

Since software is distributed in units called *file packages*, these packages must be analysed, scripted, and tested before they are handed over. They must include the development of a regression capability. A package in this context can represent a single file, a set of related files of an entire application. Whether or not the package requires a system reboot must be defined. Each file is ‘committed’ in its own right, and so must be self-consistent.

Software Distribution operates in a hierarchical manner. A package can be installed on any Windows NT server, and then distributed from that server to the network resources it serves. Packages are passed first to a local Tivoli server. This distributes it to the target environment (for example outlets), using the distribution-related hardware and network inventory to determine the distribution targets and network routes. The distribution process takes account of the network bandwidth available.

Within an outlet, the Gateway PC forwards the package to the other counter PCs. This considerably reduces the potential load on the network. The distribution can be fine-tuned to exploit a designated amount of bandwidth at each level of the hierarchy.

The Tivoli *Commit* operation is used on-line, to run the installation routine on the outlet counter PC in unattended mode. The package is re-verified before installation takes place. This routine creates a regression capability, for example taking before-looks of the NT registry, copies of replaced DLLs and so on. Installation is unattended, to avoid relying on (non-existent) technical support at the target site. In order to support this process the rules defined in sections C.3.4.1 and C.3.4.2 must be followed.

Software changes are introduced during the early morning, when the Riposte Desktop and services are shutdown. The shutdown and restart are controlled by Tivoli. Because re-booting counter PCs requires intervention from the Post Master in terms of the Post Master Logon Process (POLO) (described in section 6.3.3, *Reboot,* of the main document) such re-boots are kept to a minimum.

The rate of change of software can have performance implications, and so the rate at which it is predicted to change must be established during the *Performance Review* so that it can be modelled and monitored.

C.3.4.1 Definition of software packaging rules

The definition of each software product or product upgrade must be defined, in terms of:

- Its constituent files.
- Scripts to perform the installation (and removal) of the product, see section C.3.4.2 for the rules governing scripts.
- Criteria by which it can be asserted that a software product is installed (sometimes called a software signature).
- A naming scheme for identifying the product.

The Hand-over documentation required is described in section C.4.1 and the Application Software Package Definition in section C.4.2.

Packages must be design to ensure that they can be installed remotely, without any impact on other packages. Large package cause distribution issues. Accordingly, the target maximum size is 0.5Mb.

All changes to applications must be supplied as deltas and must provide some persistent signature as to their existence and co-exist with the target platform environment.

If there are any issues with any packaging rules, they must be raised during the *System Management Review.*

C.3.4.2 Rules for defining installation scripts

Each script provided must:

- Allow the software product or product upgrade to be distributed to one or more end systems.
- Be capable of activation on the destination system(s), including supporting the execution of any necessary data translation functions within the scripts. All existing data structures must continue to be supported during the migration period, when more than one version of a package can exist within the estate at the same time.
- Be capable of supporting reversion to a previous version of a software product.
- Support the deletion of a software product on one or more end systems.

C.3.4.3 Core and non-core software

The strategy for distribution of software is to distribute all software to the outlets, and to use Implementation Reference Data to control its use.

C.3.4.4 Software pilots

It is possible for pilot purposes to distribute software to a selected number of outlets.

Should a software pilot be a requirement then the issue must be raised during the *System Management Review.*

C.3.4.5 Software and Reference Data interdependencies

Business Reference Data is temporal in nature and so can be distributed in advance of the new version of software that uses it. Implementation Reference Data is application specific and so can be distributed in advance of the new application that uses it.

Not all counters will receive the new version of application software at the same time, so any dependencies on the order of distribution of counter and agent packages must be included in the dependencies definitions, see section C.4.3.

C.4 Application Configuration

The Application Configuration is the first phase at which each application enters the system management domain. At this point, all initial discussions and reviews need to have been completed and a definition of the characteristics of the application established. The ease with which an application can be implemented, distributed, and supported depends on the quality of the documentation that describes its behaviour.

The documentation that needs to be provided includes:

- Hand-over Note that defines the packages being handed over, see section C.4.1.
- Application Software Package Definitions that defines the components being handed over in each package, see section C.4.2.
- Definition of all application dependencies, see section C.4.3.
- Application Security Model, see section C.4.4.
- Application Performance Model, see section C.4.5.
- Description of the application being delivered, see section C.4.6.
- Documentation needed to support the application, see section C.4.7.
- Training documentation, see section 0.

C.4.1 Hand-over Documentation

The supplier of a new application must provide a Hand-over Note that allows the hand-over to be recorded and defines all the Products and Items lodged in Configuration Management that relate to the application being handed over.

The table below summarises the items to be included:

Hand-over Note

Item	***Description***
Supplier	The name of the supplier making the hand-over.
Hand-over Identifier	Uniquely identifies the particular hand-over.
Hand-over Description	Description of the contents of the deliverable.
Owner	Name of the person responsible for the hand-over.

Physical media/Delivery location	Media to be labelled with Supplier ID, Hand-over Description, sequence number of the media. The type of media that is acceptable is diskette or CD-ROM.
Components	Details of all components in the deliverable.

Table C-4 Hand-over Note contents

These definitions, for which there is a Package Definition, see section C.4.2, must include all software and Reference Data Work Packages, and Build and Installation Scripts. The scope and content of each package must be agreed at the *System Management Review*.

C.4.2 Application Software Package Definitions

Each software package must have a unique identity and name each component within a package. The documentation used to define building the application is summarised in Table C-5, below. It includes the scripts that support the build, which must be supplied as separate work package definitions.

Item	***Description/sub-item***	
Name	Name by which the item will be known.	
Type	Optional, but if supplied will form part of the physical name.	
Version	Version by which the item will be known.	
Components	All the components that are used to build the deliverable.	
	Component Name	As specified to CM, but without a suffix.
	Component Version	The version to be used in the build.
	Component Type	(Any file suffix.)
Related Change Proposals	List of all Change Proposals (CPs) incorporated. The definition of the content of numbered changes needs to be maintained for integration, system test and support purposes.	
Related test observations	List of all faults (PinICLs) for which a solution has been provided. The definition of all faults resolved needs to be maintained for integration, system test and support purposes.	

Products serviced	List of products that are to be serviced by the deliverable. Installation details are required for each entry.

Table C-5 Contents of application software package build instructions

C.4.3 Application Dependencies Definitions

Software developed to operate in the Horizon environment has impact across horizontal platforms and vertical 'slices' as described in the OPS and TMS Architecture documents. To be able to achieve such system integration all dependencies both horizontal and vertical must be identified.

The dependencies may be on other OPS or TMS components and between application components themselves. The main dependencies that must be identified are:

- OPS hardware including configurations, connectivity and usage is described in section 8, *Physical Architecture*, in the TMS Architecture document.
- OPS software including Windows NT, Riposte Desktop and software libraries and registry settings.
- TMS hardware including configurations and usage at both counters and correspondence server levels as described in section 8, *Physical Architecture* in the TMS Architecture document.
- TMS software including Riposte Message Server and software libraries version and registry settings.
- Interfaces to and from the TMS layer in terms of physical characteristics and recovery mechanisms.
- Interfaces to the security software used in the OPS and TMS domains.

There are also potentially dependencies on existing services, including:

- Audit service.
- Archive service.
- Help Desk.
- Reference Data Management services.

All these dependencies must be identified, and the impact analysed and quantified during the *Dependency Review* process.

The definitions that result from the *Dependency Review* must identify all document, software and hardware dependencies, as illustrated in Table C-6.

Item	*Description*
Input documents	A complete list of the documents that were used for the production of this product.
Software dependencies	Definition of other software needed to support the integration of the deliverable.
Hardware dependencies	Definition of any hardware required to support the integration of the deliverable.
Service dependencies	Definition any processes or NT services that may impact successful start-up or shut-down.
Data dependencies	Definition of all Reference Data objects used.

Table C-6 Dependencies definitions

Should a new application require additional support from the Horizon System Help Desk, then the impact of this support must be discussed with ICL Pathway during the *System Management Review*.

If a separate Help Desk is established, then any feed of information from the ICL Pathway system must be agreed in advance during the *System Management Review*.

Should a new application need to support HTML Help screens as well as the standard bubble Help, then the impact on the Performance Model must be discussed during the *Performance Review* and the support implication during the *System Management Review*.

C.4.4 Application Security Model

To ensure that access is controlled, controls are applied to the application functionality. The Security Model that applies to a specific application must be covered during the *Security Review* where:

- Any cryptographic and key management requirements must be identified, see Appendix B, *Cryptography and Key Management*.
- Any access control rules at the counter must be established, see section * .
- Any access control rules that the application needs to police must be established, see section C.4.4.2.

C.4.4.1 Access control at the outlet

Access Control is applied at a resource level. It means verifying that a particular user has the right to access a given resource in a particular way. Access control is defined in terms of *Roles*, each of which defines a number of functions that a user can use. Effective access control depends on having a clear definition of the roles and responsibilities of everyone who needs some form of access to the system. A user may be allowed to use several different roles, which may or may not provide overlapping sets of functions. Authorisation is the process of determining whether a user has permission to use a particular role, whether this is formally defined or not.

C.4.4.1.1 Start-up process

Control of access to the facilities within the OPS on each counter PC within an outlet is by means of the Post Master Logon (POLO) process. A smart card is used to authenticate that the Post Master is allowed to access to the counter PC.

C.4.4.1.2 Log on process

After the POLO process has been completed, users logon to the Desktop in one of the roles defined for an outlet. The access control mechanism for applications within the OPS is provided by facilities in the Riposte Desktop software.

C.4.4.1.3 User roles

The roles that have been agreed with POCL to provide access control to the Desktop at an outlet are:

- **Post Office Roles**, including Post Office Manager (Manager and Supervisor), Counter Clerks, (Clerk) and Auditor, (Auditor).
- **Operational Roles**, which provide the means to control the Horizon systems during normal running (Emergency Manager).
- **Support Roles**, such as engineers (Engineer) and support (Support, Migrate, Setup).

The access rights of each type of use are defined in persistent objects. Users with Post Office roles have passwords that have to be changed regularly. All other users have to use a ‘one-time’ password that requires a call to the Horizon Help Desk.

> Any implications of these rules on a new application must be raised during the *Security Review.*

C.4.4.2 Application access control

The Desktop itself is a hierarchic menu driven system, and this feature is exploited to allow only certain roles access to certain parts of the menu structure. This is achieved by establishing access rights to buttons within the hierarchy.

Applications are accessed from a particular level in the hierarchy, and are responsible for checking that they are being accessed only by those roles authorised so to do.

For each application, the roles by which it can be used must be defined, and in turn, the application must check that it has been called by an allowable role.

> The definition of these details must be provided as part of the Hand-over Note and discussed during the *Security Review* process so that it can be incorporated in the *Menu Hierarchy* document.

C.4.5 Application Performance Model

A new application will cause some effect on the volume of transactions and hence the performance of the Horizon system.

> The Performance Model for an application must be established during the *Performance Review*. The topics that must be addressed and documented in the model are:

- Application performance characteristics, from the point of view of the volumetric requirements as identified in section C.4.5.1.
- Usage of persistent objects, bearing in mind the issues identified in section C.4.5.2.
- Usage of messages, bearing in mind the issues identified in section C.4.5.3.
- Usage of indices, bearing in mind the issues identified in section C.4.5.4.
- Impact of the use of any online transactions, with reference to the potential performance impact identified in section C.4.5.5.

C.4.5.1 Performance characteristics of application

For a new application, the volumetric characteristics must be established during the *Performance review* so that they can be included in the Performance Model prior to the implementation of the application. The model can then be monitored when the application is system tested and when it is released in the live estate, see section C.6.3.

The capacity of each component must be modelled by platform as well as the predicted volume of data passing across the interfaces between components such as agents, WAN lines and the application. During System Tests and live running, the capacities are tested against the declared volumes for these components.

The following are required for each Business Level Transaction in the new application:

1. The number of transactions for each hour of the week (i.e. the values for 24 hours a day, 7 days a week) in the system in a typical week.
2. The number of transactions for each hour of the week in the system in a seasonal peak.
3. The number of transactions for each hour of the week in the pre-Christmas peak.
4. The distribution of transactions versus the size of outlets.
5. When seasonal peaks, other than Christmas, will occur.
6. The details of the transaction including Riposte messages produced, any interactions with the data centre, response time requirements etc.
7. Resource usage (processor, disc capacity, disc i/o, memory, LAN usage etc) for every platform that the application uses (e.g. counter, agent, host).
8. The volume and frequency of Reference Data changes.

If any part of the application runs in the data centre (e.g. an agent) the following performance statistics need to be established:

- The volume of transactions that the application will support.
- The number and type of RPC calls made against other components of the system (e.g. RPC calls by an agent to the correspondence server).
- The elapsed time for batch processes.

These volumes must be formally agreed both in the Performance Model and as a contract change during the *Performance Review* because they will be the basis of any cost that ICL Pathway incurs should a new application require it to increase the capacity of the system.

C.4.5.2 Persistent object usage

By their very nature, persistent objects should not change very often. The performance implications at the Correspondence Server level are severe if they are created or changed frequently.

Any usage at all of persistent objects, except for that defined in section 5 of the main document, must be agreed with ICL Pathway during the *Performance Review* and documented as part of the Performance Model. Wherever possible an alternative approach using transient messages will be proposed.

C.4.5.3 Message usage

Because of the volume of messages currently processed by the Horizon system, care must be taken to create the minimum number of additional messages consistent with the requirement of the application. Message expiry periods should be kept as short as possible consistent with the business requirements and potential for communications outages.

There is a default expiry period that will be used if the application does not set an explicit value. ICL Pathway reserves the right for operational reasons to change the default expiry period without notice for performance purposes, but will not reduce it such that the requirement to hold the data for two complete previous POCL Outlet Accounting Periods (normally seven days each) is compromised. So, it is important that Business specific values are used wherever possible.

The volume and size of all messages must be discussed during the *Performance Review* so that they can be incorporated into the Performance Model. The content and structure of all messages that flow across interfaces must be discussed during the *System Management Review* and documented in the Application System Description.

The expiry period for each message created by an application must be agreed with ICL Pathway during the *Performance Review* so that its anticipated effect can be established using the Performance Model, and its actual effect measured on the live estate.

C.4.5.4 Use of indices

Indices are only to be used for truly random access in performance critical operations. The way indices are created is discussed in section 4 of the *TMS Architecture Specification.*

C.4.5.4.1 Correspondence servers indices

Because of the impact on performance caused by the use of indices at this layer, the only indices that may be used at the Correspondence Server layer are those created by ICL Pathway.

This results in the following rules being applied to indices created and used on the correspondence servers:

- Applications must not create indexed messages or persistent objects.
- Indices must not be used to find all objects of a particular type. (The Riposte Scan function is very efficient for this purpose.)
- Applications must not destroy any indices.

C.4.5.4.2 Counter indices

The following rules apply to indices created and used on counters:

- Applications must only create indices if absolutely needed for random access.
- Indices must not be used to find all objects of a particular type. (The Riposte Scan function is very efficient for this purpose.)
- Applications must not destroy any indices.

> The indices created and used by the application must be established during the *Performance Review* so that they can be included in the Performance Model. This is of particular importance if new indices are required by counter applications, as a check must be made to ensure that they have no effect on the performance of other applications.

C.4.5.5 Co-ordination across outlets

Due to the nature of the distributed system, it is possible for a new application to introduce additional implicit synchronisations across multiple outlets. These synchronisations may be acceptable if they do not impact the data centre.

An example of a scenario that would have a significant performance impact would be if a new application did additional processing at the end of the day at a different time from the existing End of Day processing and required an on-line transaction to be enacted at this time.

> Any elements of the design of a new application that have implicit synchronisation implications must be raised during the *Performance Review* so that the impact can be evaluated.

C.4.6 Application System Description

The Application System Description defines the overall system design and is a key document in terms of its impact on the successful management of the system when implemented. This document must ensure that:

- All interfaces and the data that flows across them are identified.
- All interactions with the clerk are identified.
- All operational aspects are identified, including use of a Help Desk to resolve clerk concerns.
- Business Process flows are identified to enable Counter Procedures to be created.

The contents of this document must be discussed during the *System Management Review*.

Should the application use a smart card, then any issue in this area must be raised and resolved during the *SmartMan Review*, see Appendix A, *SmartMan Interface*.

C.4.7 Application Support Documentation

The Application System Description defines the overall design of each new application. The Application Support Document defines the operational aspects of the new system.

There are a number of aspects that must be addressed in this support documentation . These include:

- Any event recorded by the application must be agreed and documented, see section C.4.7.1.
- Operational aspects such as how to start and stop the application must be agreed and documented, see section C.4.7.2.
- Configuration management aspects of the operational system must be agreed and documented, see section C.4.7.3.
- Error handling and diagnostic support must be agreed and documented, see section C.4.7.4.
- Archiving and audit aspects, see section C.4.7.5.

These operational aspects must be raised during the *System Management Review*.

C.4.7.1 Event management

Event management is used to record information about the behaviour of the applications present in each outlet. It is the responsibility of each application to record in the NT Application Event Log file, those events that can have a major impact on the use of the application.

For each event the severity of the event must be defined, as must the action to be taken once the event has occurred. Because of the distributed nature of the solution, ICL Pathway has a filter mechanism that allows specific events to be forwarded to the centre for action. Since all support is remote, the application must be self-managing wherever possible, and where this is not possible there must be an agreed support procedure.

> There must be agreement with ICL Pathway during the *System Management Review* on the impact and resolution of all events and each event must be documented in the Application Support document.

There are rules that each application must follow with regard to the use of events. These are:

- Use only the NT Application event log.
- Identify all source names used.
- Document all events, their severity, and suggested recovery actions. These include system errors, application errors, time-outs, and thresholds exceeded.
- Ensure that the application correctly uses NT event types (i.e. errors, warnings, and information). Events must include a unique indication of the application's name and version.
- Document security related events or security enforcing events, i.e. that they affect or are reporting on the security integrity of the application and platform.
- Indication of the likely event volumes for normal running.
- Avoid floods of events through duplicates. The applications should treat these as inherent application state changes, and fall back to periodic error reporting on the state.

> Any potential use of the NT Security Event Log must be agreed at the *Security Review*.

C.4.7.2 Operational process control

If the application is a standalone process outside the Desktop, the Application Support document must:

- Document if it may be safely started and stopped. This may be by an API or may involve the use of the specific executables. If an API is provided then it must be synchronous and atomic, i.e. only return when all, or no, processes are started or stopped. The definition of each must be included in the components identified in the application configuration and dependencies documentation. Any peripheral interactions must also be agreed and documented.

- Document any dependencies to other processes or NT services that may affect their successful start up or shut down. These must also be included in the dependencies documentation.
- Document whether they can be (safely) restarted automatically on failure.

If the application is implemented as an NT service, the Application Support Document must:

- Document all service names used.
- Document any dependencies to other processes or NT services that may affect their successful start up, or shut down.
- Document whether they can be (safely) restarted automatically on failure.

If a service cannot be safely restarted, then the implication of this and any manual intervention required must be raised at the *System Management Review*.

C.4.7.3 Configuration management

For any configuration data that is not held as persistent objects in the message store, the Applications Support document must:

- Document any generic configurations, the APIs for setting the parameters and the underlying footprint on the NT registry, if used as the repository for these parameters.
- Document when changes to these parameters take effect. This must either be when the parameters are updated or by invoking another API. The necessity to restart the application is strongly deprecated.
- Document any dependencies on the configuration of other applications/NT subsystems and indicate the synchronisation of these dependencies.

The same documentation must be provided for all configuration items within TMS and OPS.

The use of all parameter settings must be discussed as part of the *System Management Review*.

C.4.7.4 Error handling and diagnostic support

All counters are remotely managed and have no onsite IT experienced support yet must be continuously available. The application design needs to be aware of this and obey engineering principles that support this environment. They include:

- The application must never silently fail, that is wherever possible it must record the reason for failure, even though the failure may not be made visible to the clerk, for fraud prevention reasons.
- The application must provide diagnostic facilities that can be remotely managed. The preferred model is:
  - An API is provided to switch on/off specific diagnostics. Such changes shall be immediately actioned by the application, or a separate API is provided to make the application recognise the new values. Applications must not need to be restarted.
  - Such diagnostics shall go to one or more diagnostic files. All files must be cyclic, whether disc or memory resident.
  - The diagnostic files shall be in English text format, or if encoded in other ways (e.g. binary, local language) then there must be facilities to transform the data into English text format at source.
- Applications must never rely on an operating system reboot or application restart to free dynamic resources such as memory, semaphores, temporary disc storage.
- Applications must successfully manage transitions across time changes, such as in BST. At a minimum their behaviour shall be documented in conditions where the time changes (e.g. if the application relies on any time based recovery) and they must never fail.
- Applications must provide recovery from all conditions. Ideally, this is always automated, but if not, documentation must be provided to describe the recovery action. Recovery conditions must cover an uncontrolled system shut down (e.g. power loss) and therefore an application must design out (or minimise) exposure to critical sections in the code.

The principle being use to deal with error recovery and diagnostic support must be discussed and agreed during the *System Management Review*, and the outcome documented in the Application Support Document.

C.4.7.5 Archiving and audit

Every message within the message store has an expiry period as one of its attributes. This is the number of days that the message will stay on-line in the message store. It may be set explicitly by the application that creates the message, or it may default to the system default that is set as a Riposte configuration parameter.

An expired message cannot be retrieved from Riposte even if it has not yet been archived. (See the main body of the document on the temporal nature of messages.)

Archived messages are deleted at the outlet and archived by the Archive Service at the Correspondence Server level to other media and then deleted.

Should any data created by an application be required for audit purposes then this must be discussed with ICL Pathway during the *System Management Review* to establish the effect on the existing Audit Service.

The use made by the application of archive and audit facilities must be included in the Application System Description and any operational implications in the Application Support Document.

C.4.8 Application Training Documentation

For each application a training course may need to be developed and training material must be created for the Training Mode capability provided to all users in an outlet.

These two requirements must be discussed during the *System Management Review* and agreement reached on the documentation required.

C.5 System Integration

At the point in the life cycle where a new application is integrated into the rest of the Horizon system, there are a number of key constraints that must be addressed. They are:

- The way the software is packaged in terms of its components, dependencies and any security implications.
- Its relationship to existing Reference Data.
- Its need for new Reference Data.

These will have all been documented at the Application Configuration stage. This is the first point in the System Management life cycle when errors can be raised on the delivered software and Reference Data. These errors will be registered and progressed through the ICL fault management system PinICL.

The deliverables at this stage are therefore:

- Build instructions that define the parameters needed to support the ICL Pathway platform build process.
- Application software itself.
- Any Reference Data specific to the application.
- For any fix for an error identified, specific details of the change.

C.5.1 Build instructions

ICL Pathway has an automated method for building platforms for test purposes. The definitions provided for a new application must be sufficient to enable this process to be extended to include the new application.

The details required must be identified and discussed during the *System Management Review.*

C.5.2 Application software packages

Each application package must be handed over with the appropriate Hand-over Note, see section C.4.1. If an error is identified in the software supplied, then a fix will be needed that identifies the error, and the PinICL number or numbers involved.

C.5.3 Reference Data packages

Each application must be handed over with test versions of any application specific Reference Data to allow the application to be integrated into the Horizon system and the total system tested. If an error is identified in the data supplied, then a fix will be needed that identifies the error and the PinICL number or numbers involved.

C.5.4 Fixes

For any error to be resolved, information must be provided that allows the validity of the fix to be established. The information required is summarised in the following table:

Item	*Description*
Files to be delivered	Identify the files, whether new or replacement, extent of change.
Installation Instructions	Installation details including any process dependencies.
Additional testing needed	Definition of the regression tests needed to prove that the fix is successful.
Known Problems	Identify any known issues.
Regression capability	Identify how the fix can be removed after it has been applied.
Dependencies	Desktop state (up or down) for a counter application.
Batch files to be run	For each file the following is required: • Name of batch file. • Failure conditions and how they are recognised. • What regression capability is included and how it is actioned. • What rollback action is included. • What tidy up action is included. • Any testing instructions. • Any tools needed.

Table C-7 Fix information

C.6 System Test

It is at this point in the life cycle that the application is tested in an end to end system, with the same configuration as in the live estate. Software packaging is an important element as this is the first point at which the live configuration is tested. Security is also an important element, because it may have operational implications. The way the system is used in the outlet, as well as the support capability of the application and its performance against predictions, are also tested at this point.

Errors may be raised on any of these items. The deliverables for this stage are therefore:

- Test scripts needed to prove that the functionality in an end-to-end context.
- Operating instructions for users at the counter.
- Training course and Training Mode material that can be evaluated against the application software.

C.6.1 Test scripts

These are needed to prove that the functionality provided by the application functions as predicted in the Horizon environment. Test Data Scripts must be provided that demonstrate the acceptability of the software. There must be a series of scripts to prove that:

- All interfaces support the data predicted in the interface definitions in the Application System Description.
- All operational elements of the application show the characteristics identified in the Application System Description and the Application Support Documentation.
- All events, error handling and diagnostic capability identified in the Application Support Documentation are provided and operate as predicted.
- Application interface and behaviour as presented to the clerk reacts in the way predicted in the User Operating Instructions.
- All dependencies function as predicted in the Application Dependencies Definition.
- Security components function as predicted by the Security Model.
- Application provides the performance characteristics predicted by the Performance Model.
- Training Mode for the application provides the facilities defined in the Application Training Documentation.

C.6.2 User Operating Procedures

These must cover the way the dialogue is conducted with the user in normal, error, and contingency modes.

The way the dialogue is conducted with the clerk has implications for the Counter Procedures and must be discussed and evaluated during a *Process Review* conducted during the System Test activity.

C.6.3 Performance monitoring

Checks are made to ensure that the application performs as expected from the Performance Model, and that when resources are monitored the application is found to have the predicted impact on the throughput and infrastructure performance of existing applications. This includes the monitoring of resources across the live estate.

During System Test, the performance of the application against the predictions in the Performance Model will be established. ICL Pathway is responsible for the capacity management and performance monitoring of the system, including any third party applications that use the Horizon infrastructure. Consequently, the Performance Model must define all the characteristics that support capacity management and performance monitoring.

The live service is monitored for the gross behaviour of the shared components, such as CPU usage, memory usage, and disc input and output.

C.6.4 Training course

Any training course and Training Mode material developed will be evaluated against the software delivered and issues raised resolved via the normal error correction process using PinICLs.

C.6.5 Fixes

For any error that to be resolved, information must be provided that allows the validity of the fix to be established, see section C.5.4 for details.

C.7 Application Installed and Implemented

This stage covers the installation throughout the live estate and the initiation of its use by staff at each outlet. The co-ordination of the installation of the software and Reference Data components, and the user procedures and training schedule, are the key elements at this point as is the ability to regress the implementation should errors occur.

The main deliverables at this point are:

- Counter Procedures, based on the User Operating Procedures, that include agreed manual as well as application procedures.
- Training plan that identifies how users will be trained in the user of the facilities provided by the application.

Any issues identified in these areas must be addressed at a *System Management Review*.

C.7.1 Counter Procedures

The current set of Counter Procedures must be updated to include the facilities provided by a new application. These procedures include all manual and system based procedures and identify when additional support, such as calls to the Help Desk, should be invoked.

The Help Desk and support procedures must also be updated to match the advice given in the Counter Procedures.

C.7.2 Training plan

This must cover all training activities for all the users of the application, including how any Help Desk staff will be trained to answer the calls identified in the Counter Procedures.

C.7.3 Fixes

For any error that to be resolved, information must be provided that allows the validity of the fix to be established, see section C.5.4 for details.

C.8 System Supported

After the application has been introduced into the live estate, the support of the application becomes significant as does its actual performance and the way any error fixes are introduced into the estate. The impact of the training the users receive becomes apparent in the number of calls to the Help Desk.

The key elements at this point are:

- The support needed to resolve issues identified with application functionality.
- Training received by the users and their use of Training Mode.
- Performance of the application when used by users at the outlet.
- How easily diagnostic information can be obtained and issues resolved.
- How easily upgrades can be introduced when new functionality is introduced.

C.8.1 Application support

In a distributed estate, it is important that diagnostic information can be recorded by applications in files held on the counter PC that can be interrogated centrally.

The level and kind of diagnostics depends on the application involved and must be agreed with ICL Pathway during the *System Management Review* prior to hand-over of any work packages as it can significantly affect the support resource requirement.

C.8.1.1 Resource monitoring

Applications should not assume that there is any external resource monitoring in place.

If resources require monitoring, the application must provide a self-administering routine that takes appropriate action should the resource exceeds its parameters and this must be raised at the *Performance Review*.

C.8.1.2 Version numbers

It is recommended that applications define version information for each DLL in the standard NT properties field belonging to the DLL. Furthermore, it is recommended that when an end user application is first loaded, it writes version information to the NT event log, and also to an application specific diagnostic file.

C.8.2 Training

During live operation the training identified in the Training Plan is delivered. The effectiveness of the use of training Mode will be reflected in Help Desk calls. The effectiveness of both kinds of training is continuously monitored by ICL Pathway, in terms of the impact on Help Desk calls.

C.8.3 Performance monitoring

During System Test the performance of the application will have been tested against the predictions in the Performance Model. In live running the performance will be monitored.

C.8.4 Upgrades

For any upgrade that is to be delivered, information must be provided that allows the validity of the upgrade to be established in the same way as fixes are handled, see section C.5.4 for details.

C.8.5 Fixes

For any fix to be delivered, information must be provided that allows the validity of the fix to be established, see section C.5.4 for details.