

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

Document Title:	REVISIONS TO THE TESTING & INTEGRATION APPROACH FOR PATHWAY RELEASE CSR+
Document Type:	Strategy
Abstract:	<p>This document defines the proposals for revising the strategic approach to be adopted for the testing and integration of Pathway products for Horizon/POCL, for delivery at Release CSR+. Its scope is the full band-width of existing Testing and Integration activities and covers: Changes in practice which have evolved during the course of Release 2 (now known as CSR); Changes proposed to better reflect the circumstances at Release CSR+; Lessons learnt from the testing of Release 2; Revisions reflecting changes in emphasis resulting from the recent Pathway reorganisation.</p> <p>It should be read in conjunction with the existing documentation set and in particular: General Testing Policy [1]; Testing and Integration Strategy [2]; Revisions to the Testing & Integration Approach for Release 2.0 [3]. The changes described in this document overlay those described in [3]. That is, where they cover different ground they can be taken as being cumulative in their effect, and where they cover the same ground this document can be taken as superseding the material in [3].</p>
Status:	Approved
Distribution:	<p>Development Directorate Management Team: Terry Austin, Peter Jeram, Gill Jackson, Ian Morrison Quality Manager – David Groom Programmes Director – Mike Coombs Programme Office Manager – Graham Chatten Author – John Hunt Acceptance Managers – John Dicks, Dave Holingsworth</p> <p>Pathway Library</p> <p>Horizon Testing – Richard Gaze Horizon Acceptance – Min Burdett, Bob Booth</p> <p>Horizon Library</p>
Author:	J Hunt
Approval Authority: Signature/Date:	Terry Austin, Development Director
Quality Authority: Signature/Date:	David Groom, Quality Manager
Customer Authority: Signature/Date:	Richard Gaze, Horizon Testing Manager

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010

Version: 2.0

Date: 18/11/99

DOCUMENT HISTORY

Version	Date	Reason
0.1	17/03/99	Initial draft – not issued – framework
0.2	22/03/99	Interim draft – not issued – following initial workshops
0.3	31/03/99	Interim draft – not issued – following further workshops
0.4	09/04/99	Interim draft – not issued – consolidation
0.5	06/07/99	Interim draft – not issued – reflecting contract / organisation changes
0.6	08/07/99	1 st issue for initial review
0.7	07/08/99	Interim draft – not issued – applying comments received from initial review
0.8	16/08/99	Interim draft – not issued – including agreed Joint Testing/Business Testing
0.9	19/08/99	2 nd issue for final internal review
1.0	21/09/99	Pathway Approved Issue
1.1	22/10/99	Issue for Horizon review - Incorporates changes per AI298 Resolution Plan [4]
1.2	13/11/99	Issue for Horizon review – applying comments from review of version 1.1
2.0	18/11/99	Issue as approved both by Pathway & Horizon

CHANGES FROM LAST ISSUE

Ref.	Change
Admin	Updated header to version 2.0 , 18/11/99; amended cover to show status as approved: amended document history, changes from last issue, changes forecast; added BRTS to associated documents as [5]
7.2	Amended wording as requested by Horizon to remove ambiguity
17.1	Brought section on Scripting up to date as discussed and agreed with Horizon, to reflect current position for CSR+
11	Amended as agreed to reflect indicative status of this section and to include reference to intended coverage of procedural tests.

CHANGES FORECAST

Change	Target Issue
none	

ASSOCIATED DOCUMENTS

Ref.	Library Ref.	Title	Source
[1]	VI/POL001	General Testing Policy	Pathway
[2]	VI/STR001	Testing and Integration Strategy	Pathway
[3]	VI/STR006	Revisions to the Testing & Integration Approach for Pathway Release 2	Pathway
[4]	CR/ACD/298	Acceptance Incident 298 – Resolution Plan (In particular, this reference is to section 5.5 of version 0.8 of this document, which section is also referred to as the Testing Policy document, not to be confused with the General Testing Policy [1] listed above)	Pathway
[5]	VI/STR/023	CSR+ Business Release Test Strategy	Pathway

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

ABBREVIATIONS & GLOSSARY OF TERMS

The definition of any abbreviations and terms used in this document are generally to be found in the Testing and Integration Strategy [2] and so are not repeated here.

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

CONTENTS

1. INTRODUCTION
2. MANAGEMENT SUMMARY
3. SCOPE OF TESTING
4. HIGH LEVEL TEST OBJECTIVES
5. DEVELOPMENT TEST
6. PRODUCT ACCEPTANCE TEST
7. PRODUCT INTEGRATION TEST
8. SYSTEM TEST
9. INTEGRATION TEST
10. JOINT BUSINESS TEST
11. MODEL OFFICE TEST
12. LIVE TRIAL
13. MAINTENANCE & REGRESSION TESTING
14. EXTERNAL CERTIFICATION
15. TEST ENVIRONMENTS
16. TEST AUTOMATION
17. TEST SCRIPTING
18. ACCEPTANCE TRIALS

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

1. INTRODUCTION

1.1 Background

The overall strategic approach for testing within ICL Pathway was agreed with the Authorities in September 1996 and is described in two principal documents - the General Testing Policy [1], and the Testing and Integration Strategy [2].

This approach calls for the production of a release specific test strategy for each new business release, a Business Release Test Strategy (BRTS). For release 2, a document similar to this one was produced, 'Revisions to the Testing & Integration Approach for Pathway Release 2' [3], and this also served as the BRTS for Release 2.

As the testing activities for Release 2 draw to a close and those for Release CSR+ commence, a further review of these strategies has been conducted, looking at the actual performance and effectiveness of the test activities at Release 2, examining the difficulties encountered, and taking stock of the current situation. There are lessons to be learnt, evolutions in the terms and practices used which need to be reflected, and changes in circumstance which must be taken into account.

This document summarises the proposals for change arising from the findings of that review. It also reflects changes in emphasis expected from the recent reorganisation of Pathway's Systems and Programmes directorates, with the formation of the new Development Directorate.

In addition it incorporates the improvements agreed in the Resolution Plan for AI298 [4].

1.2 Context

As with its predecessor [3] the structure of this document closely follows the general structure of the Testing and Integration Strategy [2]. This should make it easier both to review in the first place, and also easier then to overlay the necessary changes on the original document, and to produce the associated family of strategy documents for Release CSR+ from their existing equivalents for Release 1 and 2.

1.3 Underlying Approach for Release CSR+

When applying these proposals to the Strategy documents and plans for Release CSR+, it is important to remember the fundamental underlying elements of the release, and these will need to be reinforced in the subordinate test strategies for Release CSR+ when they are produced. These elements include:

- It is a mixed release (both infrastructure and business functionality). This brings particular challenges for the testing domain. In particular the various inter-dependencies between the different product sets must be properly recognised and reflected in the planning for the release accordingly.
- It is not a 'green-field' release, but rather it is introduced on top of the existing release which will already be rolled out across a significant user base. The consequent migration activities are therefore particularly sensitive. From a testing perspective, migration must be treated as just as much a part of the release as say a new area of business functionality. It must be catered for throughout the testing lifecycle.
- The counter upgrade for CSR+ will need to be conducted entirely down the wire, to a large population of outlets (several thousand), within a realistic timeframe and without significant disruption to business. It is possible that this upgrade may need to be phased over time, and so the technical viability of a mixed version regime must be validated.

REVISIONS TO THE TESTING & INTEGRATION APPROACH FOR PATHWAY RELEASE CSR+

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

- Two important and substantial infrastructure systems are introduced in the security domain – VPN and KMS. Each will require special attention.
- It introduces much wider use of Smart Card technology, involving interfaces with new 3rd party technology suppliers. This may present difficulties in the area of test data generation. An alternative tactical approach may be required to deal with this problem in the development testing area. It will also demand stringent security restrictions during testing, to protect the security elements involved in the technology, and to protect the IPR of Pathway's products where collaborative tests are run with third party technology suppliers, which may be potential competitors. These third party suppliers may also have IPR concerns of their own which may need to be factored into the planning of tests in these areas.
- AP clients are taken on directly at this release, and the existing AP client base is migrated, transferring their current feeds from the POCL HAPS system over to the new Pathway system. This client migration and the client take on process each require specific testing.
- This will be the first release where the actual target live data centre will not be available for use as a test environment (as it will already be in use in operating the live system). The testing coverage will therefore need to include specific validation of the build and release process, as the programme will no longer be able to rely on the safety net of using a build-in-situ process as employed on previous releases.
- Contractual Acceptance is brought to a conclusion at the end of Live Trial, before Release CSR+ goes live, and so does not feature as a test stage as it did for Release 2. No Acceptance Trials will need to be supported by the test programme.
- The Live Trial will take place at Release 2 and so will no longer apply for Release CSR+ or beyond.
- The position regarding Joint Testing Has been discussed at length between Pathway and Horizon, and these agreements have been substantially revised.

1.4 Next Steps

With the exception of the changes relating to the Resolution Plan for AI298 [4], the proposed changes are already approved by Pathway (at version 1.0 of this document), and are being adopted in the plans for CSR+.

This version of the document, which now incorporates the changes relating to the Resolution Plan for AI298 [4], must be formally reviewed by and agreed by Horizon from two perspectives:

- That it is a satisfactory revision for CSR+ of the existing baselined approach which was adopted for CSR and was documented in its predecessor [3], which is a CCD
- That it satisfies the related actions in the Resolution Plan for AI298 [4]

Any comments arising from the review(s) will be applied at version 2.0 of this document, which will then be the agreed definitive version.

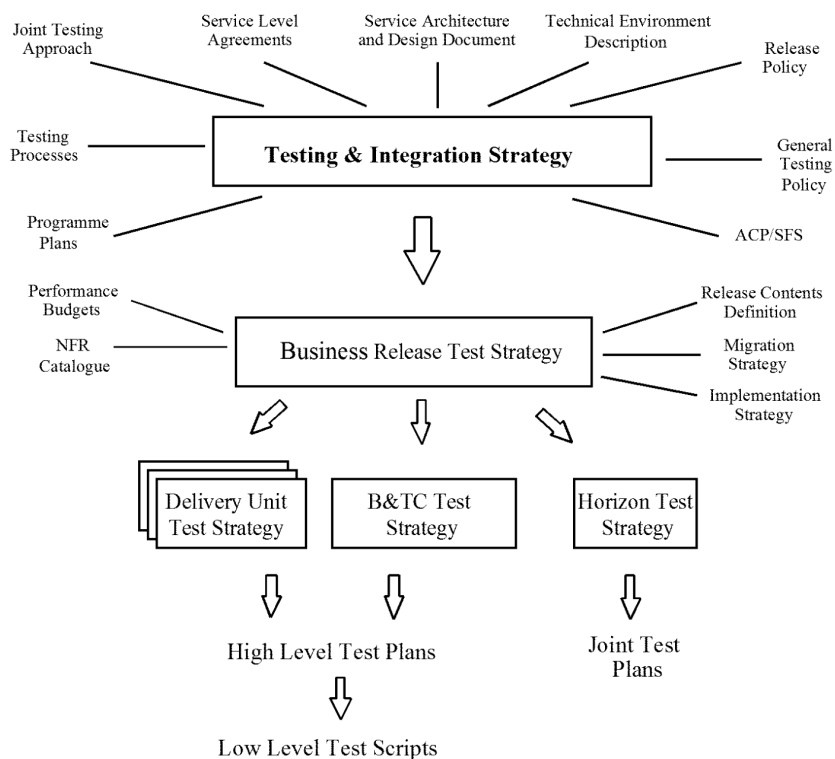
This all needs to be accomplished within the timetable laid down for it in the Resolution Plan for AI298 [4], to complete by 24/11/99.

The appropriate CCN can then be processed accordingly to make version 2.0 a CCD.

1.5 Documentation Map

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

The following diagram is an updated version of that given in section 1.3 of the Testing and Integration Strategy [2], reflecting the changes proposed in this document.



1.6 Source Documentation

As the diagram suggests, there is a great deal of key source documentation which underpins the test preparation phase and which is critical to setting the correct scope and coverage of the testing to be performed. It is taken as a pre-requisite for this testing approach that all the key source documents are complete and available for Release CSR+ sufficiently early to support the test preparation phase. These include: SADD; TED; RCD; NFR Catalogue; Performance Budgets; Migration Strategy; Implementation Strategy; SFS.

2. MANAGEMENT SUMMARY

2.1 The overall scope of testing as described in the original documents remains, except in that Live Trial is removed, and Technical Evaluation (prototyping and design feedback activities) is introduced. However, changes are proposed in the manner in which many of these objectives are to be achieved for Release CSR+. These arise for a number of reasons: lessons learned from testing of Release 2; special circumstances pertaining to Release CSR+; changes in Pathway's organisation; natural evolutions reflecting the stage the programme has reached with Release CSR+.

2.2 The principal changes proposed are:

- a) Introduce a new activity (not strictly a stage of testing) – Technical Evaluation – comprising prototyping and design feedback activities. It is intended to provide information required by the Technical Design Authority or the Delivery Units, by conducting appropriate informal trial runs of early software releases and hardware platforms, to allow the design and development of the related products to be completed prior to their entry into the later test stages, and so reduce the level of disruption to the later test stages that would otherwise result from consequent late changes. This would be under the control of the Technical Design Authority, with the runs being conducted by a purpose built team within the Business and Technical Conformance area.
- b) Reduce costs by generally increasing the use of Code Reviews.
- c) Reduce costs by generally increasing the depth, coverage, and formality of Module Tests.
- d) Reduce costs by generally increasing the depth, coverage, and formality of Link Tests.
- e) Reintroduce Product Acceptance Test, as an equivalent activity to Link Test, but for third party developments, to be owned by the Delivery Units.
- f) Use Link Test/Product Acceptance Test as the proving engine for environment builds in PIT.
- g) Further develop Technical Integration, and its interface with CM, the Delivery Units, and B&TC, to address its outstanding objectives in the following areas:
 - Configuration Settings.
 - Single Sourcing.
 - Platform Definitions and Build Scripts.
 - Environment Shells.
 - Baselines and Deltas.
- h) Extend the remit of Technical Integration to cover the management of and responsibility for the building of the Live and Support environments, via OSD/CS where appropriate.
- i) Reposition System Test, to be owned by the Delivery Units. Broaden the remit of System Test to encompass Infrastructure Systems on an equal footing with Business Applications, and to encompass the System Validation objectives previously part of Technical Test, such as Functional Load Tests, use of Infrastructure Services, specific Non-Functional Requirements of the system, etc.
- j) Reposition DIT, removing it from the Integration Test stage, to be owned by the Delivery Units, and reaffirm it as a strictly bi-lateral activity. Adopt a flexible implementation regarding end to

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

end data flows and physical data transport, appropriate to the circumstances pertaining to the particular interface.

- k) Adopt a flexible mode of working across the transition from Link Test to System Test, with close co-operative working between the Delivery Units and PIT.
- l) Remove Technical & Security Test and Business integration Test, effectively merging them to form a new test stage – Conformance Test, owned by B&TC.
- m) Remove Model Office Rehearsals, effectively replacing it with Release Test, owned by B&TC.
- n) Reduce the scope of Joint Testing, shifting the emphasis away from test execution and results analysis, and concentrating on test preparation.
- o) Remove End to End Interface Test and Model Office Test and effectively replace them with Pre-Event Testing and User Confidence Testing, based strongly on the '6 pack tests' which evolved toward the end of the user testing for Release 2.
- p) Remove the Live Trial stage as it was concluded at Release 2.
- q) Remove the relationship with Acceptance Trials, as they were concluded at Release 2.
- r) Improve the defect removal efficiency, and so reduce the number of avoidable incidents arising in the Live Service, by adopting the measures agreed in the Resolution Plan for AI298 [4], namely:
 - Extend as objectives/checklists as appropriate for Code Reviews, Unit Test, System Test, and Conformance Test, specifically to better trap the classes of defect identified in the analysis of incidents referred to in the Resolution Plan for AI298 [4] for Stability/Performance and Usability/Robustness.
 - Adopt the principles of the EPOSS Defensive Test exercise which was conducted for CSR, on a wider basis for CSR+ (i.e. not just for EPOSS but more generally across the counter applications), within the Conformance Test stage
 - Work with POCL to determine and agree an appropriate alternative for CSR+ to the Live Trial activity which was conducted for CSR.

3. SCOPE OF TESTING

- 3.1 The overall scope of testing described in the original documents has not changed, other than in two respects. First, that the Live Trial stage, which is to be conducted at Release 2, will no longer apply at release CSR+ and beyond. Second, that prototyping and other design feedback activities, which are conducted by making informal test runs, need to be formally recognised and planned for, rather than allowing them to take place by default, which at Release 2 often led to them being conducted later in the lifecycle than is ideal, and as a consequence was expensive for Pathway. This has been dubbed the Technical Evaluation activity at CSR+.
- 3.2 There have however been a number of evolutions during the course of Release 2 testing which need to be formally reflected, such as the natural gravitation of DIT away from Integration Test and toward System Test, the replacement of the former E2E and MOR/MOT activities with Pre-Event testing and a User Confidence Trial, and revisions to the Joint Testing agreements.
- 3.3 There have also of course been a number of other lessons learnt, both during the course of Release 2 testing, and during Live Trial running, which are reflected accordingly. The most significant of these are:
- that the separation of Technical Testing caused a polarisation of objectives which obstructed progress in many respects. It was a mistake which this document proposes is corrected for testing of Release CSR+.
 - The adoption of more comprehensive and earlier defensive tests for the counter applications, focussing on exception handling, negative testing, and destructive testing
- 3.4 Pathway has recently reorganised the Systems and Programmes directorates to improve its ability to operate parallel streams of work and so increase the workflow capacity. This reorganisation resulted in the formation of the new Development Directorate and is accompanied by a number of related shifts in emphasis to the testing of Release CSR+ and these are also reflected in the proposed changes.
- 3.5 It is also worth highlighting the likely involvement of new third party players representing POCL's smart card technology suppliers, such as SMS. Early discussions with SMS indicate that they may expect us to participate in an additional test activity with them, utilising their 'test bench' facilities.
- 3.6 There has been some confusion over the term Unit Test. In previous strategy documents this term has been used to embrace Module Test and Link Test, whereas the online standards used it to describe Module Test and defined Link Test separately. For consistency it is reiterated that Unit Test embraces both Module Test and Link Test activities, and the online standards will be updated accordingly.
- 3.7 All these different changes are described in more detail under the appropriate sections below.

4. HIGH LEVEL TEST OBJECTIVES

- 4.1 All the existing high level testing objectives remain unchanged, with the exception of Live Trial which is now removed. (Of course, where the means by which each test objective is achieved is being revised, then the specified terms under which the objective will be assessed may also require revision, and this should be reflected in the respective detailed strategy documents. For example, MOR/MOT objectives where previously couched in terms of utilising a 'laboratory office' whereas this may not be appropriate for the replacement UCT activity.)
- 4.2 A small number of additional ones are introduced, such as with the introduction of the Prototyping and design feedback activities.
- 4.3 Some are reorganised and expanded, such as with Integration Test.
- 4.4 They are all described in more detail under the appropriate sections below.

5. DEVELOPMENT TESTING

In the reorganisation of the Systems and Programmes directorates a series of Delivery Units are being established, based around related product sets which together form end to end systems, and taking these product sets through the development lifecycle from Design through to System Test. So, where the hand over point between development testing and independent testing used to be following the Unit Test stage, it is now on completion of System Test. This will allow the new Delivery Units to test and integrate their product sets on an end to end basis before hand over to independent test, in a manner that simply was not possible for the previous development organisation.

Previously the testing activities conducted within the development domain consisted of Module Test and Link Test, which together made up the Unit Test stage. For Release CSR+ it is proposed that the scope of testing here is somewhat broadened as follows.

(Note, the scope of testing conducted by the Delivery Units also includes Direct Interface Test, and System Test, described under section 8 below.)

5.1 Prototyping and Design Feedback – Technical Evaluation

During the course of Release 2 testing, a pattern has emerged in certain of the later testing activities, most notably in the Technical Testing areas. In many such areas it was not initially possible to conduct and complete formal tests as planned, as it became necessary to carry out a stream of iterative informal tests. These were effectively prototyping some of the detailed system parameters and configuration settings, and providing other similar design feedback information.

This was a costly and disruptive process. These later test activities were unavoidably delayed as a consequence. They did not form the ideal vehicle for providing this information. It was too late in the lifecycle, and a number of late changes were the inevitable result, which in turn is disruptive and expensive.

It is proposed that for Release CSR+ the need for such information is formally recognised, and the necessary prototyping and design feedback activities are planned into the programme from the outset.

These have been referred to as Technical Evaluation and will be under the direction of the newly introduced Technical Design Authority (TDA). They may be initiated directly by the TDA, or via them on behalf of the Delivery Units. They will generally be conducted by new purpose built team within the Business and Technical Conformance (B&TC) group. The implementation details are yet to be finalised, and will be documented by the TDA and the B&TC in due course.

The high level objective of this activity is:

to provide the necessary information, by conducting appropriate informal trial runs of early software releases and hardware platforms, to allow the design and development of the related products to be completed prior to their entry into the later test stages, and so reduce the level of disruption to the later test stages that would otherwise result from consequent late changes.

5.2 Code Review

In general it is believed that Release CSR+ would benefit from greater levels and more formal conduct of Code Reviews than was the norm for Release 2. This would serve to reduce the number of incidents encountered in Module and Link Test, and generally improve the quality and consistency of the code at the earliest point in the lifecycle, and so reduce Pathway's costs. The format, content and coverage of these code reviews should be agreed between the Delivery Units and the Quality Assurance Manager.

Further, in accordance with the Resolution Plan for AI298 [4] :

- Where reasonably necessary the Code Review Checklists for counter applications will be extended to include confirmation of sufficient and consistent exception handling.

5.3 Unit Test

There has been some confusion regarding Unit Test. The previous testing strategy documents have used the term to refer to the test stage within the development arena, encompassing both Module Test and Link Test. Whereas, the online standards have used it as a replacement term for Module Test and define Link Test separately. This document reiterates the original position, and the online standards will in due course be updated to reflect this also.

5.3.1 Module Test

Module Test applies to in-house developments only. It is expected that any third party supplied products will have been subject to equivalent levels of testing, in accordance with the General Testing Policy [1].

In general it is believed that Release CSR+ would benefit from greater levels and more formal conduct of Module Test than was the norm for Release 2. The observation in Release 2 testing was that too many of the defects uncovered by the later test stages could more economically have been trapped in earlier test stages. This suggests that the balance between development testing and independent testing was still a little out, and that a greater proportion of development testing would help reduce costs and time scales overall.

More specifically, Module Tests will be formally documented and planned for, such that they form auditable and re-runable test packs. They will deliberately be planned to operate as autonomously as possible, to remove potential planning bottlenecks, by stubbing out extraneous interfaces and exploiting the formal APIs for the products concerned. Greater use of test harnesses and test drivers would facilitate this, and would also simplify wider use of test automation techniques and dramatically reduce the cost of regression testing. Similarly, the tests will be planned to operate on synthetic data, further removing unnecessary interdependencies.

It will always be possible to conduct Module Test on the same platform used to develop the module.

As the programme moves into the position of having a fully operational and fully deployed system in the field, work on subsequent releases, such as Release CSR+, demands that regression testing features ever more strongly in the development testing domain. To this end, the target for path coverage in Module Test should now be increased to achieve as close to complete coverage as is practicable. (It is accepted that there will always be exceptional circumstances in certain areas, which prevent full coverage being achieved. Such cases will require justifying before they can be sanctioned.)

Infrastructure products should be treated on equal terms. They are not exempt from this testing requirement.

Further, in accordance with the Resolution Plan for AI298 [4] :

- The BUSY.EXE tool will be deployed and used on Module Test environments to trap memory usage defects.

5.3.2 Link Test

Link Test applies to in-house developments only. It is expected that any third party supplied products will have been subject to equivalent levels of testing, in accordance with the General Testing Policy [1]. (However, see reference to Link Test under Product Acceptance Test at 5.4 below.)

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

In general it is believed that Release CSR+ would benefit from greater levels and more formal conduct of Link Test than was the norm for Release 2. The observation in Release 2 testing was that too many of the defects uncovered by the environment build process, and by the later test stages could more economically have been trapped by Link Test. This suggests that a greater proportion of Link Testing would help reduce costs and time scales overall.

(Such changes have been mooted in the past, but were preoccupied with extending the scope of Link Test to embrace cross platform linkages. They were not adopted because at the time this would have involved increased hardware costs in making sufficient suitable test rigs available to the development areas, and this eroded the cost benefits of the changes. However, these changes now do not on the whole require cross platform involvement, and exceptionally where it is required (see below), following the recent reorganisation, the new Delivery Units do now have the full System Test environments available to them to satisfy this.)

More specifically, Link Tests will be formally documented and planned for, such that they form auditable and re-runable test packs. They will deliberately be planned to operate as autonomously as possible, to remove potential planning bottlenecks, by stubbing out extraneous interfaces and exploiting the formal APIs for the products concerned. Greater use of test harnesses and test drivers would facilitate this, and would also simplify wider use of test automation techniques and dramatically reduce the cost of regression testing. Similarly, the tests will be planned to operate on synthetic data, further removing unnecessary interdependencies.

For the purposes of Link Test a product set (e.g. APS Host) will generally be confined entirely within a single platform (e.g. Host Layer). Multiple product sets normally co-operate across platforms forming a system (e.g. APS). It is not expected that Link Test for a given product set will extend beyond the limits of that product set, and so will not generally need to extend beyond the immediate platform boundary, but the cross platform APIs will be isolated (e.g. stubbed out) and checked. Hence it will normally be possible to conduct the Link Test on the same platform used to develop the product set concerned. (There will always be exceptions to this general rule. For example, VPN and KMS by their nature do not lend themselves to this approach. It is likely that link testing of these products would be planned on an end to end basis and so will span multiple platforms. Under such circumstances Link Test would probably need to be run on the System Test rig, and the distinction between Link Test and System Test will be somewhat arbitrary.)

Infrastructure products should be treated on equal terms. They are not exempt from this testing requirement.

Once a product set has successfully passed Link Test, before it can be deployed in System Test it will need to be formally handed over via CM and built into an appropriate controlled test environment. Part of this hand over will be the auditable test results and re-runable test packs for Module Test and Link Test.

The Delivery Unit concerned will proactively hand their products over into the Technical Integration area, operating their Link Tests on the newly built rig. This will achieve three further objectives: to demonstrate that the product set was ready for build; and to accelerate diagnosis of problems during the build process (as Link Test has already been run successfully, then any failures encountered in this proving process should be related to build problems only, and not to unknown product states); and to confirm that that the rig has been correctly built and is suitable for use in System Test.

The 'rig' referred to here may be the TI rig, or preferably the target rig for the subsequent System Test activity, depending on the circumstances/availability prevailing at the time.

Further, in accordance with the Resolution Plan for AI298 [4] :

- Where reasonably necessary the Objectives and Review Checklists for Link Test, for the counter applications, will be extended to include coverage and confirmation of sufficient and consistent exception handling, and provision of adequate interlocking to preserve and protect the serial dependencies inherent in the VB runtime environment

- The BUSY.EXE tool will be deployed and used on Link Test environments to trap memory usage defects.

5.4 Product Acceptance Test

At Release 1 Product Acceptance Test was a test stage in its own right, conducted by the independent testing area, prior to System Test. It was not very successful. At Release 2 it was combined with PIT in an attempt to overcome the difficulties. Whilst PIT has been generally very successful, the Product Acceptance Test objectives were still not always achieved. This left Pathway exposed to potential poor product quality from its 3rd party suppliers. Fortunately this did not become a serious problem, but it could be in the future if not properly addressed.

In retrospect there were still two things going wrong here. First, the proving scripts used tended to be cross platform in nature (often originating from System Test material), and so did not isolate the product sets. Second, the ownership was in the wrong place – neither PIT nor the testing organisation were equipped to ‘accept’ a product set from a 3rd party supplier.

The Delivery Units are best placed to perform this activity. It is proposed that Product Acceptance Test be conducted as the equivalent of Link Test, but for such third party deliverables. This would keep it at the right level (within the confines of a platform). It would also keep it within the right area of expertise (the area which is delivering the other related product sets), and would retain the right level of ownership. Added benefits are that it would provide a very early opportunity for feedback to our suppliers, would provide an in-built regression test facility for external product sets, and would remove planning bottlenecks, as each product set would be independent of others, not having to wait until a full system was assembled before they could be accepted.

All the original test objectives of the Product Acceptance Test stage defined for Release 1 are carried forward into this test activity.

PIT could then treat third party products in the same way as in-house products, with the Delivery Units concerned using their Product Acceptance Test packs in the same way as their Link Test Packs (see 5.3.2 above).

Where appropriate it may be beneficial for a Delivery Unit to arrange for the bulk of Product Acceptance Testing to be performed on the third party’s environment.

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

6. PRODUCT ACCEPTANCE TEST

- 6.1 It is proposed that Product Acceptance Test will no longer form a part of Technical Integration, nor return to its Release 1 format as a discrete test stage, but will rather proceed as one of the test activities within the new Delivery Units, being conducted in much the same manner as Link Test, but for third party products only. (See section 5.3.2 and 5.4.)

ICL Pathway **REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

7. PRODUCT INTEGRATION TEST

7.1 Product Integration Test (PIT) was introduced as a new test activity at Release 2. It has on the whole been very successful, dramatically improving the environment build position over that experienced at Release 1. However, certain of the objectives set for the new area were not fully achieved, and the following proposals are made to address these items.

7.2 Configuration Settings

At Release 2 (and earlier), a number of problems have been experienced regarding both specific and general configuration settings and parameters.

One example is where particular configuration settings are given default values by their suppliers. These values have in very many cases been left to default until very late in the lifecycle before more appropriate values have been established for them.

Another example is where the configuration settings are of an environment sensitive nature (e.g. database sizes may vary, being very small for many test areas, of a significant size for some other test areas, and full sized for live). Typically these have not been managed within CM, but rather by the different environment build teams, hand stitching these settings as they go. The result being that there is no certain configuration in use for any given environment at any given time. They simply 'evolve'. Another side effect is that for those configurable items concerned, all the other settings, which may not be environment sensitive, tend to get tarred with the same brush. This casts doubt on whether or not the 'real' settings are in fact being tested.

One more example is where settings are consciously changed, but because they do not feature in the CM definition there is no record of the change within the CM system, only in the build areas which carried them out.

These anomalies are each on their own fairly small and may seem unimportant. However, taken cumulatively and in combination they can have a devastating effect on the stability and reliability of the environments concerned. This in turn has a severe knock-on impact on testing. The following proposals are aimed at avoiding some of these problems and so reducing their impact on testing.

It is proposed that settings should not be left to default, but should be set explicitly. Where a particular supplier's default settings are believed to be appropriate, they should nonetheless still be specified to be set explicitly, and the values should be documented in the Platform Design and annotated that they were the same as the default values, such that any changes to them are highlighted at future versions.

The new Technical Design Authority will confirm the appropriateness of such defaults. Where insufficient data is available to do this, the necessary prototypes will be run to determine whether or not they are appropriate under the circumstances pertaining to the Pathway solution.

7.3 Single Sourcing

It is proposed that where particular configuration settings are environment sensitive (i.e. have to take different values from environment to environment) a single value should be treated as the 'master'. This would normally be the one intended for use in the Live environment. The configuration item(s) concerned will be delivered as a 'single source' with this 'master' setting(s). Each environment variant would be identified and delivered as an associated transformation CI(s) for use in the build process. In this way there is no confusion about what the 'real' value should be, and there is confidence that all the other attributes of the build which are not environment sensitive are being used as specified for Live, and are not being hand stitched each time. The process is known as 'single sourcing'.

REVISIONS TO THE TESTING & INTEGRATION APPROACH FOR PATHWAY RELEASE CSR+

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

The combination of these two proposals (7.2 & 7.3) would mean that all settings used in the build process are as specified for the CI within CM and endorsed by Design. Hence the problem relating to undocumented changes of configuration settings would dissolve – they are all within CM.

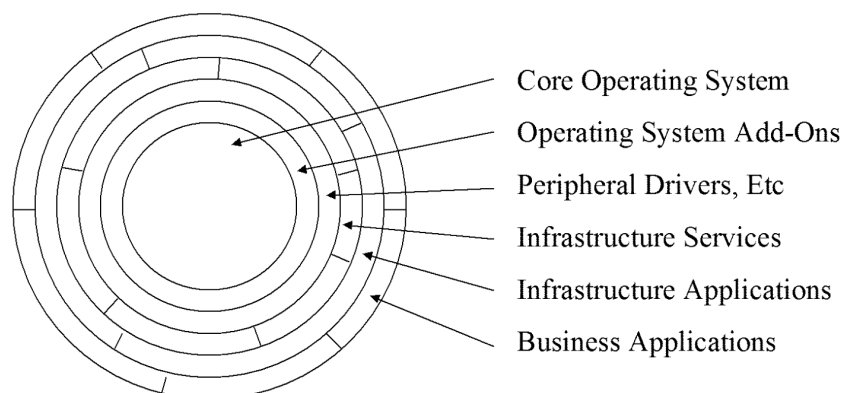
7.4 Platform Definitions & Build Scripts

A new area within the Technical Design Authority – ‘Platforms’ – will be responsible for collating information across the TDA and the Delivery Units to form a comprehensive definition (design) for each platform type. This will then enable the Technical Integration area to formalise the process of developing Build Scripts, rather than relying on trial and error. It is expected that PIT will build test environments from these Build Scripts, and maintain them accordingly.

7.5 Environment Shells

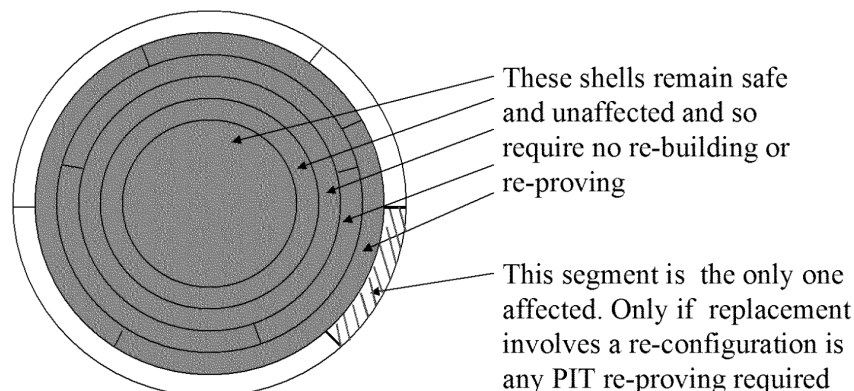
The concept of shell products was introduced when PIT was first formed for release 2, but it has not been possible during Release 2 to fully implement them. It is important for Release CSR+ that this position is rectified if the costs and elapsed time involved are to be reduced. In the past it has too often been necessary to conduct extensive re-building and re-proving of environments for relatively minor changes. The shell concept is one where the components making up a particular platform type are structured and managed in such a way as to eliminate much of this, by clearly delineating between independent groups of software, so that it can be seen more easily whether or not (and to what extent) re-building and re-proving is necessary.

The approach can be visualised as an onion skin model, with the range of diverse software types being arranged as successive layers. These can be further divided within a layer into segments. The objective always being to group software on the platform into easily manageable and separately defined sets – effectively ‘productising’ them.



(The above diagram is intended to be purely illustrative. The actual structure required must be determined jointly between Configuration Management, Technical Integration, Business & Technical Conformance, and the Delivery Units. Tivoli delivery must also be taken into account.)

This then enables the whole environment to be viewed as a number of shells which can be independently manipulated accordingly. For example, if a particular delta only involves a business application delivery, then it can be seen that nothing beyond that segment can possibly require a re-build, and if the delivery concerned is a simple replacement and not a re-configuration, then there will be no need to re-prove the configuration in PIT either. The potential savings are enormous.



7.6 Baselines & Deltas

The observation at Release 2 was that in general re-baselining of environment configurations (consolidating previous baseline with deltas made since that baseline) was performed too frequently, whilst the deltas themselves (increments) were generally on the large size. This coupled with the fact that environment shells were not available gave rise to a number of difficulties in managing the environments both from a build and a test perspective. 'Fast Tracks' became the order of the day.

The model proposed for the future is to reduce the frequency of re-baselining, to exploit the shell concept, and to reduce the size (increase the granularity) of the deltas applied. Both PIT and the test areas would operate against a 'rolling baseline' of the last real baseline plus the deltas applied since it was cast. The shells would avoid much of the need for re-building and re-proving, which would make the application of deltas more flexible and more reliable, and so in turn would allow the rolling baseline to span a greater number of deltas (increments) without risk of instability. Smaller, more frequent, more reliable deltas would reduce the call for fast tracks which could then be used purely for exceptional circumstances.

The need for establishing fresh baselines (and the attendant overheads) is thus reduced. The occasion will nonetheless still periodically arise when it is convenient consolidate the position, combining all the deltas applied since the previous baseline, and start a new rolling baseline. Typically this would be as products move from one major test stage to another (e.g. on entry to System Test and Conformance Test), when commencing a Final Pass for Audit purposes, and when responsibility passes from one organisation to another (e.g. a counter baseline would be cast prior to handing the build over to Celestica).

Such management of baselines and deltas must be agreed between Configuration Management, Technical Integration, Business & Technical Conformance, and the Delivery Units, to embrace these principles and so reduce the costs and elapsed time involved.

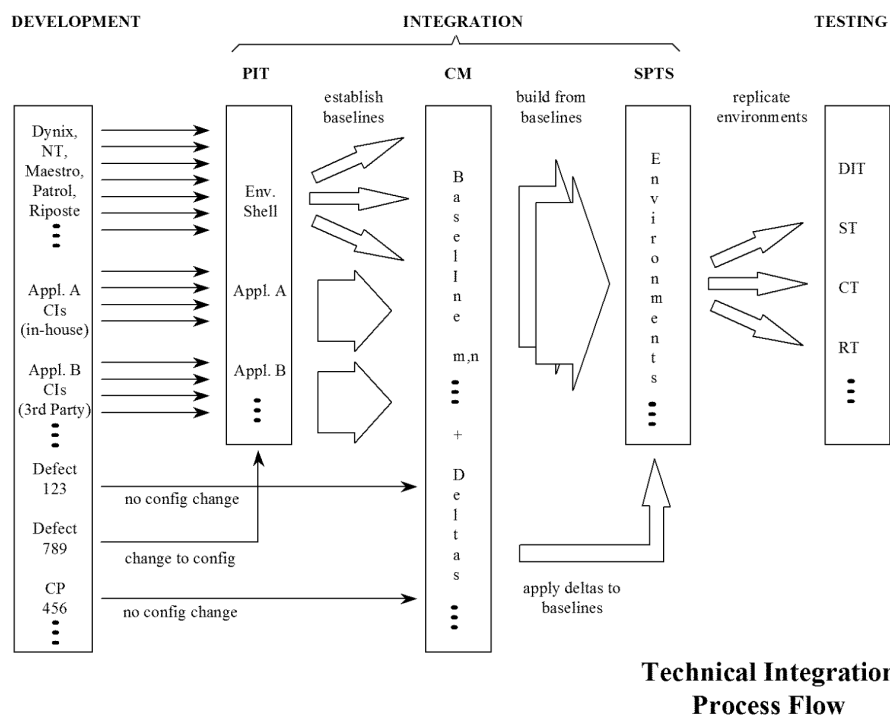
7.7 Extended Responsibility

In the past Technical Integration has been responsible for the building of Test Environments, whilst OSD has been responsible to Customer Services for the building of Live and support environments. This has led to a number of anomalies in the configuration of these respective environments. It is proposed that the remit of Technical Integration be extended to encompass both sets of environments. Whilst OSD would retain responsibility for executing the builds for Live and Support environments, this should come under the direction of Technical Integration, to ensure that all environments are built from a consistent set of Build scripts and adhere to the concept of single sourcing.

7.8 Process Flow Overview

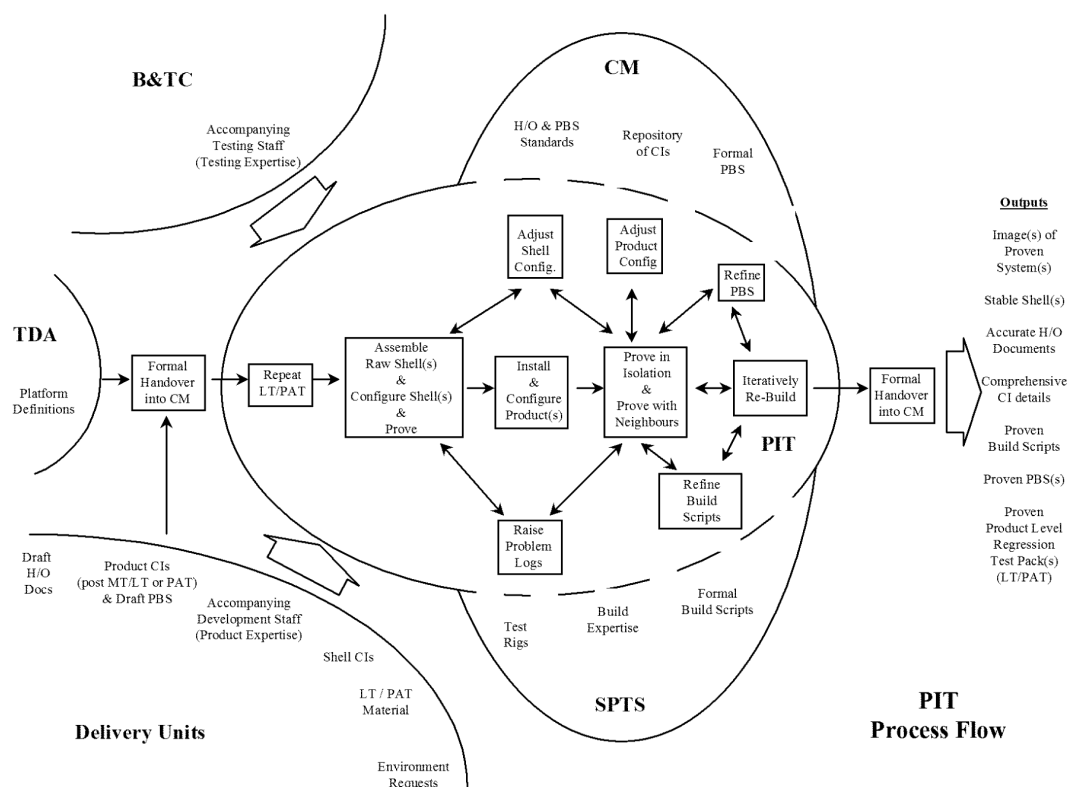
The overall process flow for the Technical Integration area, and more particularly for PIT, remain largely unchanged, and are illustrated by the following two schematics.

(Note, these schematics are not intended to imply that CM are only involved/used after PIT. On the contrary in fact, the Delivery Units hand over to PIT by lodging their products in CM, but not as a full configuration, simply as a set of CIs, to help with version control. This information is supplemented by the Platform Definitions provided by the TDA. Then through the process of integration within PIT the configuration information becomes progressively more firm as formal Build Scripts are developed, and is registered in CM accordingly, so that by the end of PIT the products are fully configured in CM and ready to build from.)



REVISIONS TO THE TESTING & INTEGRATION APPROACH FOR PATHWAY RELEASE CSR+

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99



7.9 Relationship between PIT and Delivery Units

It is proposed that once a product set has been made ready for delivery to PIT, following Link Test, or in the case of third party products, Product Acceptance Test, the delivery unit concerned would formally hand over, via CM, into PIT. As previously, it is expected that this hand over would be accompanied by appropriately experienced staff.

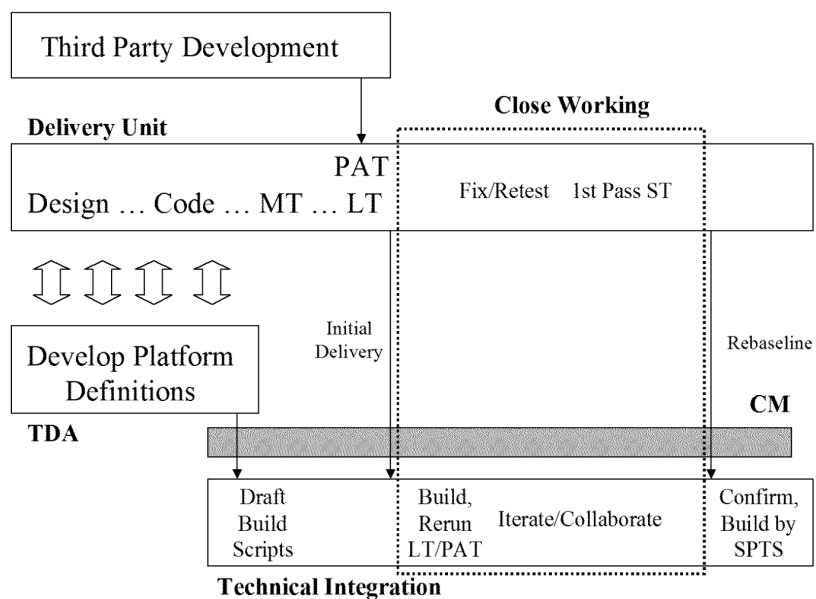
Based on the Platform Definitions provided by the TDA, PIT would build these products onto their respective platforms. PIT and the DU staff provided would then run the Link Tests or Product Acceptance Tests to confirm the status of the build. This may be an iterative process. Once confirmed the Build Scripts could be developed accordingly. This activity could take place either on a PIT rig or on the DU's System test rig. (The latter is preferable.)

If not already built on the System Test rig, this would be the next step. System Test 1st Pass could then be run. It is proposed that this is conducted in a collaborative fashion with PIT, with the minimum of formality. Once the environment, scripts, and data have been successfully stabilised, the build scripts can be finalised, and the configuration can be re-baselined in CM. PIT should confirm that this new baseline and the finalised build scripts have taken properly before passing on to SPTS, who would re-build the System test rig ready for Main Pass.

The following schematic is intended to illustrate this relationship.

REVISIONS TO THE TESTING & INTEGRATION APPROACH FOR PATHWAY RELEASE CSR+

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99



8. SYSTEM TEST

All the previous objectives of System Test are carried forward. Direct Interface Test is also now formally recognised as being more closely allied with System Test than with Integration Test. All the previous objectives of DIT are similarly carried forward. In addition the following principal changes are proposed:

8.1 Organisation

Previously both DIT and System Test were conducted by the independent test area. Following the recent reorganisation, the responsibility for both DIT and System Test now falls to the newly formed Delivery Units.

8.2 System Test Scope and Coverage

At Release 2 there was a tendency for the two main test sites, Feltham and Bracknell, to polarise to the two extremes of business oriented testing and technically oriented testing respectively. One result of this was that the scope and coverage of System Test, which was performed at Feltham, became restricted to business functionality only, with little or no coverage of infrastructure products or non-functional requirements, which in turn gravitated toward the Bracknell site under the banner of System Validation.

This polarisation is unhealthy and makes for more costly integration of the business and technical aspects of the system in the later stages of testing.

At Release CSR+ it is proposed that a conscious effort is made to redress this imbalance, by reuniting these test objectives within the new Delivery Units, extending the scope and coverage of System Test to include infrastructure components and the basic non-functional aspects of the system.

The detailed coverage intended by each Delivery Unit can be specified and agreed within their respective test strategies. In general, the following examples indicate the proposed shifts in emphasis:

- Infrastructure systems, such as Riposte, KMS and VPN, should be recognised as fully fledged systems and subject to System Test just like a business application system would be.
- System Tests should include coverage of their usage of and compliance with any general infrastructure services employed by the system concerned. For example, compression routines, digital signing facilities, encryption routines, acknowledgement agents, Maestro scheduling, general housekeeping routines, data transport mechanisms, error handling and event collection mechanisms, etc. For the avoidance of doubt, this does not extend to running System Test against fully secure test rigs. It is recognised that this would in many instances prove impractical, particularly in the management of stubs.
- System Tests should include collection of basic performance measures not previously collected in Unit Test.
- Where appropriate, System Test should encompass a functional load test. This would not be designed to stress the system, or to conduct an accurate performance evaluation, but rather to confirm that the system concerned was not disrupted by operating against a modest load. (Frequently during the course of Release 2 Performance Testing, it was discovered that the systems had basic functional problems (not performance problems) as soon as they were operated with a modest volume of data or a modest transaction throughput.
- Similarly, System test should include the operation of basic resilience and recovery features, within the bounds of the system concerned. This should include confirmation that the application/infrastructure concerned maintains data and process integrity across common failure scenarios.

-
- It is worth reiterating that confirmation of the HCI has always been within the remit of System Test
 - System Test should include coverage of the basic security features of the system concerned, in so far as they can be sensibly verified within the bounds of that system, to confirm that their expected contribution toward the overall security solution is being delivered.
 - Particular emphasis should be given to the Migration aspects of the system.

Further, in accordance with the Resolution Plan for AI298 [4] :

- Where reasonably necessary the Objectives and Review Checklists for System Test, in respect of their counter application components, will be extended to include coverage and confirmation of sufficient and consistent defensive measures within the HCI, for example guarding against abuse of the printing interface, confirming correct integration of the application with the supporting infrastructure services, and ensuring correct nesting of applications within counter sessions.
- The BUSY.EXE tool will be deployed and used on System Test environments to trap memory usage defects.

8.3 Direct Interface Test

DIT was from its inception recognised as a test stream which could be conducted in parallel with System Test. It was in fact only included within the Integration Test stage because it was by nature concerned with the integration of the Pathway systems with other external systems.

Following the recent reorganisation, with the creation of the Delivery Units, it now makes sense to sever this link with Integration Test and to place this test stream where its dependencies dictate, closely allied with System Test and under the responsibility of the Delivery Units.

One of the lessons learnt from DIT at Release 2 was that it is over ambitious to attempt multi-lateral testing at this early stage in the testing lifecycle. It is therefore proposed that DIT be limited to its original remit – purely bi-lateral.

Whilst all of the previously defined test objectives for DIT are in general carried forward, it is recognised that not all of these objectives will apply equally to all product sets. There will be exceptions where there is good reason to deviate from the norm. So, the manner in which the different Delivery Units approach DIT may differ. Similarly, the extent to which they operate their external interfaces and the respective co-operating systems may also differ. This will largely be at the discretion of the responsible Delivery Unit Manager, but these deviations will need to be formally agreed in advance with B&TC, so that everyone concerned is aware of the specific coverage intended. The differing approach/extent that may be adopted by the various Delivery Units fall into a number of categories, as follows:

- a) For an interface (such as OBCS-OBCS) which is already operating Live, and for which there are no planned changes to the interface, then DIT can be reduced to a simple regression test activity. (e.g. if each party ran simple stand-alone tests, possibly by re-running their own parts of the DIT tests from the previous release, to confirm that the data involved remained correct and so that the interface had not regressed, then there would be no need to exchange the data either manually or by the automated transport mechanism.)
- b) For an interface (such as TPS-TIP or APS-HAPS) where the communication medium (the physical link and the transport management software, e.g. FTMS) is already operating Live and it is not planned to change it, then it is reasonable to operate DIT with manual data exchange (e.g. email or floppy disk). It would be left to B&TC to confirm the fully integrated communication.

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

-
- c) For an interface (such as TPS-TIP) where there is considerable interplay with other product sets which remains visible to (even essential to) the testing of the interface, then it is unrealistic to expect the Delivery Unit to run all these interacting systems for DIT. In such cases it will not be practicable to operate DIT with full end to end flow of data. It would need to be restricted to the interface itself, with full end to end flow being left to B&T where it can operate in an integrated fashion.
 - d) For an interface (such as TPS-TIP) where there is considerable interplay with other interfaces (in this case reference data) and so it is not meaningful to operate DIT in a bi-lateral fashion with end to end data flows, then it is reasonable to restrict DIT to the interface itself, with the full end to end data flow being left to B&TC where it can be operated in a multi-lateral fashion.
 - e) For an interface (such as APS-Client or LFS-SAPADS) where the interface is new or substantially changed, and where it can be operated essentially in isolation, then it is important that DIT be operated with full end to end data flows.

It is further proposed that the coverage of DIT be re-examined, to consider the merits of including inter-system interfaces internal to Pathway. For example, an obvious candidate would be the interface with SORBUS. Any possible candidates should each be considered and included as appropriate at the discretion of the Delivery Unit Management team.

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

9. INTEGRATION TEST

9.1 Integration Test is the area predominantly affected by these proposals for change.

At Release 2 this test stage was formally broken down into a number of separately recognised test streams, as follows:

- DIT - Direct Interface Test
- T&S - Technical & Security Test
- BIT - Business Integration Test
- E2E - End to End Interface Test
- MOR - Model Office Rehearsal

Each of these streams will be dealt with in turn in the following paragraphs.

9.2 Direct Interface Test

It is proposed that DIT be removed from the Integration Test stage, recognising its affinity to System Test, under the responsibility of the newly formed Delivery Units. (See section 8.3 above.)

[The remainder of the responsibilities, described below, which were previously held by the Test & Integration group are now taken up by the newly formed Business and Technical Conformance area.]

9.3 Technical & Security Test

At Release 2 there was a tendency for the two main test sites, Feltham and Bracknell, to polarise to the two extremes of business oriented testing and technically oriented testing respectively. One result of this was that the Technical Test streams rarely included the use of business functions or representative business scenarios in their testing. Similarly, the Business Integration Test stream at Feltham rarely included the use of infrastructure products. This meant that it was not possible to derive as much benefit from these respective streams as would otherwise have been the case, and that the combined tests were pushed further down the lifecycle, with all the attendant implications.

At Release CSR+ it is proposed that the Technical & Security Test and Business integration Test streams be dissolved as separate test areas, and recombined under the new banner of **Conformance Test**.

All the previous objectives of these respective areas are carried forward into Conformance Test (except for System Validation, which it is proposed is covered within the Delivery Units, see section 8 above) with a renewed emphasis on combined tests.

Particular emphasis should be given to the Migration aspects of the system. Scenarios should be included covering the transition from CSR, through any interim increments, to CSR+, and the use of MIMAN and MIECCO and their consequential data.

With the completion of Release 2, to a very large extent the job of evaluating the non-functional aspects of the system in an isolated manner is over. The system operates as a whole. For Release CSR+ it is important that the combined objectives are addressed in a combined fashion – such that not just the performance of the system is evaluated, or the integration of the business functions is trialed, or the resilience features are rehearsed, or the security aspects probed, but rather that we also cover the performance of the security aspects and the integration of the business functions with resilience, etc.

Further, in accordance with the Resolution Plan for AI298 [4] :

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

-
- Where reasonably necessary the Objectives and Review Checklists for Conformance Test, in respect of their counter application components, will be extended to include coverage and confirmation of sufficient and consistent defensive measures within the HCI to guard against inter-system interference, for example contention and correct release of printing resources shared between applications
 - Mount a defensive test exercise (which may subsume some or all of the above bullet item if deemed more efficient and effective to do so), similar in principle to but broader in scope than the EPOSS Defensive Test exercise conducted at CSR. This exercise, running under the Conformance Test umbrella, will concentrate on the counter applications, but will not limit itself to just EPOSS. It will comprise a series of short, highly focussed test actions designed to probe the robustness and usability of the overall counter system, with particular attention to system stability under adverse usage. This exercise will include among its coverage: complex multi application scenarios, not constrained by intended user procedures; stressing of the counter printer sub-system; the effects of abuse/misuse and failure of peripherals; general abuse/misuse of the HCI (POCL could be approached to provide end users to facilitate with this aspect).
 - The BUSY.EXE tool will be deployed and used on Conformance Test environments to trap memory usage defects.

9.4 Business Integration Test

It is proposed that the BIT stream be dissolved as a separate stream, with its objectives merged with those of the T&S streams under the new banner of **Conformance Test**. (See 9.3 above.)

9.5 End to End Interface Test

E2E was an Horizon owned phase of testing introduced at Release 2. It was intended to concentrate on procedures and business integrity. It operated with end to end data flows spanning all the participating systems in the Horizon programme, with a particular emphasis on financial reconciliation, utilising highly complex scenarios. Initially this test phase had mixed success. Over time it evolved through a number of guises, culminating in a highly focussed set of test activities, dubbed the 'six pack' tests.

These latter tests proved to be extremely effective and relatively economic. However, considered overall, E2E was an expensive and time-consuming exercise. These lessons have been taken into account in Horizon's plans for their testing of Release CSR+. These are described under section 11 below.

Accordingly, from Release CSR+ the E2E test stream will cease to exist in its previous form.

9.6 Model Office Rehearsal

From Release CSR+ MOR will cease to exist in its previous form. (As described in section 11, Horizon will no longer operate a Model Office Test at CSR+, so it follows that the Model Office Rehearsal must also cease. The Pathway objectives of the MORs will be taken up in the newly formed **Release Test** described below.

At previous releases the objectives Pathway hoped to satisfy in the various Model Office Rehearsals were severely compromised. Despite numerous reconfigurations and re-positioning of these activities, addressing timing, ownership, and parallel activity, it did not prove possible to achieve the goals. The MOR remained firmly a Horizon focussed pseudo UAT. Pathway could not use them to

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

test its 'Office' (the data centres) and so in each case separate migration and implementation rehearsals had to be set up late in the day to satisfy these objectives.

At Release CSR+ it is proposed that Pathway pulls away from this previous pattern, and plans explicitly and separately to conduct its own 'model office' trialing the whole pathway system on large scale test rigs, and covering migration, implementation and operational procedures. Particular emphasis should be given to the Migration aspects of the system, running full blown rehearsals to confirm the intricate inter-dependencies involved. To remove all doubt surrounding the term Model Office, this test stream will go under the new banner of Release Test. It will run separately from and generally lagging behind Conformance Test.

10. JOINT TESTING (previously called Joint Business Test)

Throughout Release 2 testing, Pathway and Horizon continued in their efforts to implement the agreed principles of Joint Testing. In some areas this did prove extremely successful, and a great improvement over Release 1, though not without expense. In other areas progress was hampered in much the same way as that experienced previously for Release 1, with difficulties in resourcing sufficient appropriately skilled and empowered Horizon staff, and with continued misalignment of the parties objectives.

These difficulties have been discussed at great length between the parties concerned with a view to establishing the best way forward for Release CSR+. Both Pathway and Horizon agreed with the underlying principles of Joint Testing (and indeed the wider concept of Joint Working), and believed that the approach did offer significant benefits. However, the inescapable conclusion reached was that the parties simply were not suited to the approach, and that despite best intentions, the approach would continue to fail in its implementation. It was agreed that the best way forward was to salvage those areas of the approach thought to deliver the maximum overall gain for the minimum investment, and to largely abandon the remainder, substituting instead a separate Horizon controlled User Confidence Test.

As a result, it has been agreed that for Release CSR+ Joint Testing will consist of the following:

- a) A small number of appropriately skilled Horizon staff will be integrated with the Test Analysis areas within Pathway to be actively involved in the production of Pathway's test plans and scripts. The objective is to inject business knowledge directly into the test preparation process, and to provide Horizon with direct visibility of Pathway's planned test coverage.
- b) DIT will continue to operate as a joint activity (under Pathway's control) between the participating system areas, though on a strictly bi-lateral basis (see section 8.3 above).
- c) This joint interface testing will continue, on a progressively more multi-lateral basis, in the Horizon controlled Pre-Event Testing (see section 11 below).
- d) The focus of the Horizon input to the joint activities will have a strong 'business' bias, with little or no 'technical' emphasis.
- e) There is no intention to support test execution (other than DIT and PET as described above) as a joint activity, although it will be possible to support limited witnessing of Pathway test execution by Horizon staff, by prior mutual agreement.

11. MODEL OFFICE TEST

It has been agreed between Pathway and Horizon that for Release CSR+ the two parties will break away from the pattern of E2E/MOR/MOT tests previously adopted (see sections 9.5, 9.6, 10). Instead Horizon plan, and Pathway agree to support, an approach broadly as described below. It should be noted that these details are intended to be indicative and not definitive. The fine detail will be established in Horizon's test strategy documents, which will first be agreed with Pathway.

11.1 Pre-Event Testing

Taking a form similar to that of the '6 pack' tests performed toward the end of the Release 2 test programme, PET will follow on from DIT, progressively integrating the participating systems across Horizon, including Pathway. It will test business events, scenarios, and procedures, with end to data flows spanning all the participating systems. In general it is intended to proceed in 3 main phases.

- a) Bi-lateral
- b) Tri-lateral - Partial integration across the data streams
- c) Multi-lateral – fully integrated end to end data flows spanning all participating systems

PET will operate in parallel with Pathway's Conformance Test activity. It will serve as Horizon/POCL's primary measure of readiness for entry to the User Confidence Test.

PET will be planned, prepared and controlled by Horizon. It will adopt a short cycle length and concentrate on key end to end business scenarios. The principal objective of these tests will be to confirm that each of the participating systems is capable of running, that the interfaces hang together, the end to end data flows are supported correctly, and that the systems and their respective organisations are ready to mount the User Confidence Test together. A certain amount of procedural testing is also envisaged.

The test plans and scripts will be produced in advance and made available to Pathway, who will analyse the material and as far as is practicable will incorporate the key aspects of the scenarios into their Conformance Test material for ongoing regression testing purposes.

11.2 User Confidence Test

UCT will follow PET and Pathway's Conformance Test. It will serve as Horizon/POCL's primary measure of readiness for Live Release, directly informing the RAB.

It will be planned, prepared and controlled by Horizon. As the cycle length for this activity will have a direct bearing on the elapsed time required to complete the exercise, Horizon have agreed to keep it to the minimum level necessary to satisfy the underlying business data life-cycle.

The principal objective of the UCT will be to prove the referential and financial integrity of the end to end reconciliation and settlement process, and to inform the RAB. Procedural tests will also feature prominently.

Pathway will provide the necessary technical infrastructure and supporting services representing their responsibilities in the Live service.

Workshops will be held with the participating systems (including Pathway) to agree the appropriate scope and coverage of the tests, ensuring that they will both satisfy the objectives, and can be accommodated in the available time window, allowing for contingency.

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010

Version: 2.0

Date: 18/11/99

To this end it is important that much of the test material comprising the UCT will have been effectively pre-proven in earlier test stages, to provide confidence that UCT can operate successfully without planning for many iterations. Therefore the test plans and scripts will be produced in advance and made available to each of the participating systems, including Pathway, who will analyse the material and as far as is practicable will incorporate the key aspects of the scenarios into their own test stages. If this material is not available sufficiently in advance to allow for this, then the UCT is liable to be compromised.

12. LIVE TRIAL

- 12.1 The Live Trial is a phase of the operational Trial, and as such is concluded with Release 2 (now known as CSR). There is no Live Trial for Release CSR+.
- 12.2 However, it is recognised that the extended exposure provided by the Live Trial was invaluable. Certain classes of defect cannot reasonably be detected with certainty by normal testing. A small residue of defects will always remain undetected and only emerge through actual Live use. Some such defects however, whilst they may be of a minor nature in system terms, can nonetheless be very disruptive, imparting a disproportionate impact on the Live Service. It was in revealing such defects, and in restricting the extent of such impacts, that the Live Trial brought real value.

Therefore, as agreed in the Resolution Plan for AI298 [4], Pathway and POCL will work together in determining and agreeing appropriate alternative(s) to a Live Trial for CSR+, to allow each new product to be exposed to substantial Live use, but with limited business impact, for an appropriate period of time prior to general (national) release.

13. MAINTENANCE & REGRESSION TESTING

- 13.1 The objectives of this area remain unchanged. However, to remove confusion a brief description of the approach to regression testing is included below.

The test plans drawn up for each part of the system and for each stage of testing are, wherever practicable, designed to be re-runable test packs, including the necessary test scripts, test data, and expected results. These may be automated or may be run manually.

When a change is made to the system, this is either as a result of a fix being applied to address an incident (a PinICL), or following the approval of a Change Proposal (a CP). Both PinICL and CP changes are formally controlled. Part of the lifecycle of a PinICL is the planning of the testing required to satisfy it. Part of the lifecycle of a CP is the impact assessment, which includes planning of the testing required to satisfy it.

In both situations, the testing required normally breaks down into two distinct types.

- The testing of the change itself – often conducted as targeted testing, and no different to any already described in the rest of the strategy.
- The testing required to confirm that the change has not caused the rest of the system to regress (has not inadvertently introduced some unwanted side-effect) – regression testing.

The change may introduce new components, or may change existing components, or may remove existing components, or some combination of these. The targeted tests are satisfied by introducing, changing, or removing test conditions (or adjusting the expected results) relating to these components accordingly. The regression testing is satisfied by identifying the neighbouring/interfaces components surrounding the change, and in turn identifying appropriate tests (from the existing re-runable test packs) which will exercise these areas of the system. Typically this will involve scripts from each major stage of testing through the lifecycle. Also typically (and particularly for the later stages of testing) these tests will not be run discretely, but rather will be accumulated over a period of time and run in batches, for efficiency. This is all a matter of judgement.

Consider 3 examples:

- a) A minor change in a discrete application on the counter platform

Regression testing can be restricted to the application itself, and be accommodated within the Link Test area. A later run of relevant System Tests (for other purposes) could be used for further confirmation.

- b) A significant change to a host application including a schema change

Regression testing should be considered across the host application, and any others sharing that Schema, which may include related Agent applications. A combination of Link Test and System Test are likely to be required. Where the affected systems include an external interface, then DIT regression runs should also be considered. Where the affected systems have critical performance factors (or other NFRs) which may be compromised, then Conformance Tests regression runs should also be considered.

- c) Major changes to fundamental pieces of system infrastructure, highly invasive in nature

Blanket regression testing should be considered across the whole solution.

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

The extent of regression testing required is a matter of judgement, and in turn is dependent on the extent of the changes applied, and the nature of the product(s) concerned.

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

14. EXTERNAL CERTIFICATION

- 14.1 No changes are proposed in this area. The vast majority of external certification required for the Pathway solution will already have taken place for Release 2. The few remaining areas should continue as currently planned.

15. TEST ENVIRONMENTS

15.1 General.

The reliable and consistent replication of test environments has improved enormously during the course of Release 2, but still remains an area for improvement, with the potential for great savings in both costs and elapsed time. The changes proposed for PIT/CM are designed to assist here.

15.2 Data Centres

For all previous releases it has been possible to use one of the two target live data centres during testing, both to support some large scale performance tests and to support the Model Office activities. As Release 2 (now known as CSR) will be operating Live on both of these data centres, they will no longer be generally available for testing purposes in the same way.

16. TEST AUTOMATION

- 16.1 When test activity commenced on Pathway in 1996 the target technology (notably NT) outstripped the availability of test automation tools able to support it, and the product fluidity (notably Riposte data formats and Oracle Schemas) detracted from the projected benefits of automation.
- 16.2 As a result only very limited use of test automation has been made to date in Release 1 and Release 2 Testing, and this has for the most part been confined to the performance testing areas, in the Oracle/Sequent bench-marking activities at Weybridge, and in simulating transaction loads on PO counters at Bracknell and in the 20 Counter Tests at Feltham.
- 16.3 The time is now ripe to start exploiting test automation tools on a wider basis. Initially it is proposed that we should concentrate our efforts in 4 key areas:
- a) capture facilities to assist in preserving more meaningful audit trails, and to assist in diagnosis of problems, with particular emphasis on tests run at the counter.
 - b) replay and script edit support to facilitate cheaper and more extensive regression testing across a wide spectrum of functional tests (Here 'script edit support' refers to facilities which allow editing of captured material to correspond with minor changes that may need to be made to the test scripts, to keep the captured material current and avoid having to re-capture for every small change. They also permit replication and variation of captured tests to help generate loads and multiple permutations of test scenarios.)
 - c) load simulation, extending current practice, for both performance and resilience testing, and for functional load testing
 - d) network simulation, for both performance and resilience testing
- 16.4 It is further proposed that during the course of (and as an integral part of) detailed test planning for Release CSR+, any additional candidates for test automation should be identified and assessed. Where clear benefits can be projected within Release CSR+ timescales, they should be adopted accordingly. Where the benefits are longer term, they should be flagged for later study.
- 16.5 It must be recognised that test automation tools can be invasive, and so under certain conditions it may not be practicable to deploy them (e.g. where a particular test requires a fully secure environment, and the introduction of the tool compromises this position).

17. TEST SCRIPTING

- 17.1 The combined tests proposed for the new Conformance Test stream necessitated a review of the formats of HLTPs and LLTSs to investigate how the approach could best be supported. This concluded that by consistently adopting the HLTP and LLTS formats used in CSR for business oriented testing at Feltham, which are scenario based, to now cover both the business oriented and technically oriented tests, then these can be readily combined as required without any major clash in structure arising. The generic process is described at a high level in the Business Release Test Strategy for CSR+ [5], and this will be further expanded in the subordinate lower level strategies for each area.

ICL Pathway

**REVISIONS TO THE
TESTING & INTEGRATION APPROACH
FOR PATHWAY RELEASE CSR+**

Ref: VI/STR/010
Version: 2.0
Date: 18/11/99

18. ACCEPTANCE TRIALS

- 18.1 Contractual Acceptance concludes with Release 2 (now known as CSR), and so does not apply to later releases. There is therefore no requirement for preparing and conducting Acceptance Trials at Release CSR+.