

DRAFT

REPORT:

[REDACTED] Group Data Transfer

Cybernet Computer Systems Ltd

13 DECEMBER 2006

Contents

Section	Page number
1. Background	3
2. Consultant Profile	3
3. Scope of Document	4
4. Project Outline	4
5. Destination Systems	4
6. Inhouse Database Tables	5
7. User Requirements	6
8. Data Validation	8
9. System Validation/Acceptance Testing	9
10. Data Takeon Strategy	10
11. Non-Data Takeon Considerations	11
12. Summary	12

Group Data Transfer

Background

The [REDACTED] and [REDACTED] merged in October 2006. Cost savings from the merger have been identified, including moving transaction processing to a common software product, Civica's Inhouse system.

[REDACTED], [REDACTED] [REDACTED] are already Inhouse users and the decision was taken to migrate data to Inhouse from the other disparate systems in use within the new group, including [REDACTED] currently in use by [REDACTED].

[REDACTED] have decided to leave the group, but 1600 [REDACTED] properties need to be migrated to [REDACTED] system, as they are still to be managed by the new group.

Other properties currently managed on the [REDACTED] system will also need to be migrated; and these will be distributed to [REDACTED] and [REDACTED] Inhouse systems.

Consultant Profile

Gary Pigott has over 12 years experience of the Inhouse system, 4 years of which was as an Implementation Manager for the software supplier (now Civica).

Gary has performed data migrations both to (and from) Inhouse for many organisations – which have included Moat HS, Plume HA, Bourne HS, Pollards Hill HA, Manchester & District HA, East Thames Housing Group.

His skills set also includes application design and development, 4GL/4Js/Genero programming, Unix scripting, Informix, SQL Server, ASP programming, user & technical training, and project management.

Recent projects include:

Civica – development of web based Sales Order Processing system (written in ASP, HTML, Javascript – database SQL Server)

Moat Housing Group – development of Genero Sales & Marketing system; development of Inhouse-to-Contact Manager synchroniser

Affinity Sutton Group – 4Js development of bespoke programs including HB interface; migration of data from Informix SE to IDS; provision of technical training

Origin Group (St Pancaras HA) – data migration from Civica's Context (CTX) database to Orchard; development of web based (Intranet) enquiry mechanism for archived tenant, account, and repairs data (ASP/HTML/Javascript – flat file structure)

East Thames Housing Group – data migration from Inhouse to Orchard

DRAFT

Customers include :

- Civica PLC
- Moat Housing Group
- Affinity Sutton Group
- Shaftesbury Housing Group
- Origin Group
- Dominion Group
- East Thames Housing Group

Scope Of Document

This is intended to provide a technical background to the project, and provide a basis for discussion of the issues involved in the population of the destination database systems.

Other aspects of the project such as user training, operational and procedural changes, etc will also need to be considered, and although they may be mentioned are outside the scope of this document.

Project Outline

The project consists of 3 separate data transfer objectives, namely:

- i) transfer 1600 properties from [REDACTED] system to [REDACTED]
- ii) transfer 1500 properties from [REDACTED] system to [REDACTED]
- iii) transfer 500 properties from [REDACTED] system to [REDACTED]

The data to be transferred consists of property and associated data (tenants, accounts, and repairs).

The timescale for completion of this project is 31st March 2007, training and user acceptance would need to be completed before this date.

Destination Systems

The destination housing system at both [REDACTED] and [REDACTED] is Civica's Inhouse housing management system - written in 4GL (4Js), running under Unix.

The backend database for Inhouse at both sites is an Informix relational database - [REDACTED] uses Informix Dynamic Server (IDS), and [REDACTED] uses Informix Standard Engine (SE).

[REDACTED] and [REDACTED] have run-time and development licences for 4Js.

In addition, [REDACTED] also run Civica's Contact Manager (CM) system – and the destination data would also need to be uploaded to CM.

DRAFT

Inhouse Database Tables

The Inhouse system is a modular based system consisting of the following modules:

Central (CS)	property and people data
Rents (RS)	accounts, rent breakdown, and transactional data
Repairs (RP)	maintenance data
Lettings (AL)	lettings data

The data transfer (data takeon) exercise is a property based one, and as such it is assumed that lettings data will not be required – to be confirmed.

The main tables loaded (excluding lettings) are:

Estates	estates
Blocks	blocks (groups of properties)
Property	property
Stat_hist	property status history
Element	property rent details
Tenant1	tenant
Dependants	household members (including joint tenants)
Tenant2	accounts, link between tenant & property
Items	transactions
Item_audit	transaction breakdown (e.g. rent breakdown)
Scr_head & Scr_data	general scratchpad (tenant and property based notepad)
Repairs	repairs data
Rep_tasks	repairs tasks (SOR items that make up a repair)
Rep_hist	repairs history (status history)
Rep_notes	repairs notes (worknotes)
Rep_access	repairs access data (if applicable)

Optional repairs tables include:

Contractor	contractor data
------------	-----------------

Tasks & SOR item details
Task_desc

The above may need to be populated if a full repairs breakdown is required - the preferred solution would be to load historical repairs with a dummy SOR item with a description such as "see Worknote for details", and to populate the worknotes with a full description of the repair.

There may be a requirement to load additional data tables - e.g. Direct Debit (DDs) details, Gas servicing details etc.

User Requirements

A precursor to this exercise is to determine (as far as possible) the user data requirements for the data transfer, as this will be one of the main drivers for the specification of data extracts from the source system(s).

This will include decisions on the following:

- What data is to be extracted/loaded
This includes decisions such as – Will you be loading estates ? Is the estate structure still relevant ? Do you need household members? Will you be loading repairs data?

This will be constrained by limitations of the source housing system(s). Although it may be possible to combine any such data with other external data – like Gas Servicing information etc (supplementary spreadsheets, access databases etc).

- What level of data is required
For example do you want all transactions or just B/F (or Opening) balances; or do you only want the last N (tba) years worth ? Loading full transaction data may incur performance issues (see Non-Data Takeon Considerations).

The same applies to other historical data such as rent elements – do you want to load all rent elements or just the current ones.

Similarly repairs.

All or just current tenants ?

Is there data on the source system that doesn't readily fit into the Inhouse database structure – do you need additional tables/bespoke for such information, or do you want to hold such information outside of Inhouse (e.g. Intranet) ?

- Reference numbers
Are the current property, block, estate, and tenant reference numbers still relevant or do you need to re-reference to conform to destination system requirements?

Are there reference number clashes – with the existing destination data.

DRAFT

If you re-reference, do you need to populate foreign postings xref table for possible cash and other payments ? (see Non-Data Takeon Considerations).

- Estate/block structure still relevant
This also applies to other property related data such as housing management area (patch), repairs areas, company codes etc
- Rent Groups
What rent groups/debit cycles/rent calendars need to be defined for the new properties ?
- Coding structures/code changes
You are loading into existing functioning systems, it is assumed that where possible would want to retain existing system codes. How do these map to the supplied data sets, this includes things like:

- Property type/sub-type
- Housing management area/sub-area
- Repairs area/sub-area
- Company code
- Transaction types
- Rent element types
- Tenant types
- Ethnic origin codes
- Family composition codes
- Tenure types
- Payment methods

(Populated in system tables stext/attcodes)

This code mapping may require a data analysis exercise.

- Default values
Where no source data is available what defaults, if any, should be against property/tenant fields ?
- Data cleanse
What is the state of the source data – is a data cleanse exercise (manual or automatic) required ? This may include:

- Address data (address analysis/postcode validation)
- Tenant (& household) data, names and titles
- Mixed case conversion (letter production)
- Invalid dates

This cleansing may also require a data analysis exercise.

- How are errors resolved

DRAFT

In the event of errors being detected in the data transfer process (e.g. overlapping tenant accounts), who decides how these will be resolved ? Rules can be incorporated into the data upload process.

It is important to remember that the data transfer process is generally an iterative one with an initial load followed by user testing, subsequent amendments - both to the data transfer, and possibly source data extract - and numerous reloads until the user is happy with the data that has been loaded. This means that decisions made throughout may also change, and the transfer process should be flexible enough to cater for this.

Data Validation

The destination system is only as good as the data that is loaded. It is essential that the data loaded is checked, and meets user requirements. Internal resources need to be identified and planned for this process.

Definition of user data 'sign off' must be included within the project - what are the 'data check milestones' ? These would normally include some of the following:

- Number of properties/blocks/estates/tenants/accounts

- Sum of tenant balances

- Sum of transactions

- Sum of rent elements

But there are other less easily measurable data checks that are required:

- Is the quality of the data right, did you need a data cleanse ?

- Have you loaded enough information (can you answer tenant queries)

- Are the right tenants living at the right properties ?

- Are the data codes right (company code, tenure type etc) ?

- Are the right transactions against the right accounts ?

Other decisions need to be made with regard to:

- Level of data checking – everything or sample checks ?

- Responsibility for data checking

- Formalise process (data sign off sheets perhaps?)

- Are any non-standard validation reports required (is resource available to produce this)

DRAFT

This validation process would also be iterative, and will continue right up to completion of the project. This is especially important since the data is being loaded into an existing live system - so as well as checking that the 'new' data is correct we also need to ensure that the existing data is still correct.

System Validation/Acceptance Testing

Does the new system hang together ?

With the data loaded do standard system functions still work – this would include things like:

Can you query a property

Can you query a tenant

Can you query an account

Do the account transactions display correctly

Can you raise a debit, is the total debit amount correct, do the tenant balances update correctly

Can you post cash, HB, and other transactions

Can you raise DDs

Can you raise a repair (both against old & new properties, including blocks)

Do any Finance interfaces still work (do you want loaded data to go through any interface?)

Do existing reports/bespoke programs still work as expected

Can you produce a tenant statement

Can you produce arrears letters

Can you void a property, can you let a property etc

Are there any performance issues

This may be IT led, but again there should be some form of user sign off.

Data Takeon Strategy

As you are effectively loading into live systems – we would recommend the following data takeon strategy be adopted:

- create empty databases on [REDACTED] & [REDACTED]

DRAFT

- pre-populate with existing system codes (stext, attcodes)
- pre-populate with other system parameters (rent groups etc)
- backup these 'empty' databases
- convert & load the source data into these databases
- perform user data checking, initial validation, and initial acceptance testing/training on these databases (much easier to sign off new data loaded, and check routines work with new data)
- copy live databases over these 'empty' ones
- reload source data into these, resulting in live + new data
- user data checking is repeated, as is system/functionality testing
- sign off of data load completed
- performance and full functional testing, possibly including partial parallel runs
- system acceptance sign off
- repeat live (timing to coincide post completion of all Finance interfaces, preferably just prior to debit run and after a full database backup)
- remove now redundant data takeon databases (or keep as test/development systems)

Non-Data Takeon Considerations

As well as the pure data transfer based decisions and issues, there are other considerations which will include the following:

- Resourcing
Internal and external – if you need additional resources will they be available ?
- Licences
Inhouse licencing is based on numbers of properties, do these need to be increased for these extra properties (+ contingency for new properties) ?

Are extra 4Js licences required for additional users ?

- Performance
Are there associated system performance issues related to the increase in transactional data (more relevant to SE), is there sufficient capacity on the box for extra users/data ?

Will Cron timings need to be looked at due to increased run-times ? Do you want to automate more of your processing (e.g. debit runs, cash posting runs etc)

If any system upgrades (Inhouse/Unix/Informix) are planned in the near future, then we would recommend these be done prior to the data transfer, and given time to 'bed in' otherwise it may be difficult to identify the source of problems (upgrade or data transfer related).

- Training
Is there a user training requirement, and if so who will do it
- 4Js Client Software
Rollout to new user's PCs
- User set up
On Unix and Inhouse. Establishing user access levels.
- Informing affected Tenants
Do you need to write to affected tenants informing them of the change (including changes in payment procedures etc).

The same will also apply to external affected agencies (HB etc).

- Rentcards
Are new rentcards required ?
- DDs
Are new DD mandates required ?
- Payment cards

DRAFT

Will tenants need to be issued with new Giro/Payment cards? How do you ensure they pay using the new payment cards ?

If they pay using the old cards how can you ensure you get the money ? Do you need to establish formalised procedures for collecting money that gets paid to the 'old' owner (in the case of [REDACTED]).

Summary

In summary the following needs to be considered:

- Gain user buy in (user led project group, access to decision makers)
- Who will be doing the (source) data extract ?
- Who will be doing the (destination) data transfer ?
- Who decides what data is required, and other associated (user led) decisions ?
- What data is required ? Does it need to be formally specified, is sign off expected ?
- Is any data analysis required ? (may be as part of the above)
- What sign off is required ?
- Ensure appropriate resources (internal & external) are available
- Ensure non-data issues are considered, and appropriate solutions put in place