Office of Housing Assistance Contract Administration Oversight
Rental Housing Integrity Improvement Project (RHIIP)
Error Tracking Log Initiative

RHIIP Error Tracking Log (ETL) To-Be Process Model

ETL & TRACS Industry Group Meeting
March 2008

Overview

- Approach
- Where We Are Today Based on Working Group Progress
- To-Be Process Model
- Next Steps to Achieve the To-Be Model
- Summary of Benefits

Approach

Three Phases to Recommending the To-Be Model

Phase 1 - Define and document current business processes and needs for long-term improvements.

Phase 2 - Define and document requirements for the long-term solution.

Phase 3 - Define and document the solution for long-term improvements.

Using this information, HUD management will determine next steps to move directly to the Design Phase of the Long-Term Solution.

Approach

Phase 2 – Define and document requirements for long-term solution

- Analyze current business processes by conducting facilitated working sessions with HUD and MIP staff to gather additional requirements
- Define and publish a draft of the HUD System Development Methodology (SDM) Functional Requirements Document
- Conduct and host a TRACS Industry Meeting to review and refine functional requirements

Approach

Phase 3 – Define and document the solution for long-term improvements

- Conduct facilitated working sessions
- Utilize the approved and agreed-upon requirements to document the recommended long-term solution in the Solution Definition Document (recommended To-Be Process Model)
- Create and publish an Error Tracking User Guide for the recommended process model
- Update the Rent and Income Determination Quality Control Monitoring Guide to be approved by the Office of Management and Budget (OMB)

Where We Are Today Based on Working Group Progress

Current Methods of Error Tracking

- TRACS Summary and Detail Reports
- CA Vendor Software
- Voucher Administration Vendor Software
- Rent and Income Determination Quality Control Monitoring Guide – Quality Control Log based on the 2003 publication

Where We Are Today Based on Working Group Progress

Standardized Error Data

- Error Points of Discovery
- Error Types
- Error Categories
- Error Causes
- Error Resolutions

Where We Are Today Based on Working Group Progress

Error Points of Discovery – When Will Errors Be Identified?

- Voucher Review
- MOR
- Resident Contact

Where We Are Today Based on Working Group Progress

Error Types – What Errors Will Be Tracked/Reported?

- **Error Types** – what errors have the largest dollar impact on subsidy over- or under-payment?
- **Error Detail Data** – what additional error data must be captured?

Where We Are Today Based on Working Group Progress

Error Categories - How Should We Categorize the Errors?

Administrative and Component categories as outlined in the PD&R Study were too broad. They were further broken down into four lower level categories:

- **Eligibility** - household member status; ineligible expenses and/or allowances claimed, etc.
- **Calculation** - TRACS calculations differ from those reported (income, allowances, expenses); MAT records not being processed due to differences in reported figures; number of units, etc.

Where We Are Today Based on Working Group Progress

Error Categories - How Should We Categorize the Errors? (contd.)

- **Timing** - certification effective date is greater than next re-certification date; certification missing; effective date precedes move-in date; move-in before move-out, etc.

- **Reporting** - incomplete data reported: number of units reported different than contract number of units; required fields missing; invalid codes; household member listed more than once, etc.

Where We Are Today Based on Working Group Progress

Error Causes

The most common causes for errors discovered are:

- Resident Misreporting
- Resident Oversight
- Handbook Interpretation
- Handbook Error
- OA Misreporting
- OA Oversight
- OA Lack of Training – Property Management
- OA Lack of Training – Property Management Software
- OA Property Management Software System Error
- OA Property Management Software Program Error
- OA Property Management Software User Error

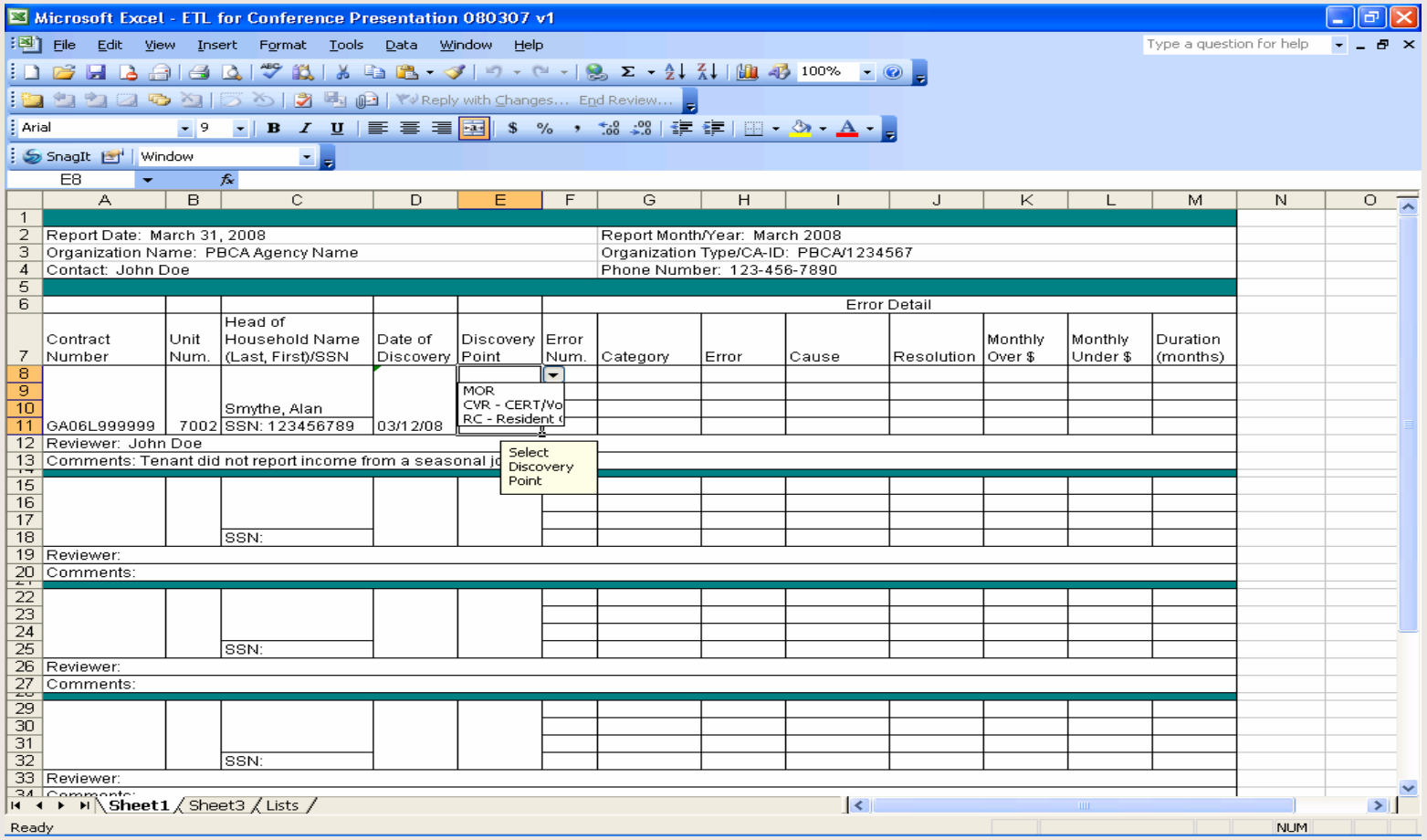
Where We Are Today Based on Working Group Progress

Error Resolutions

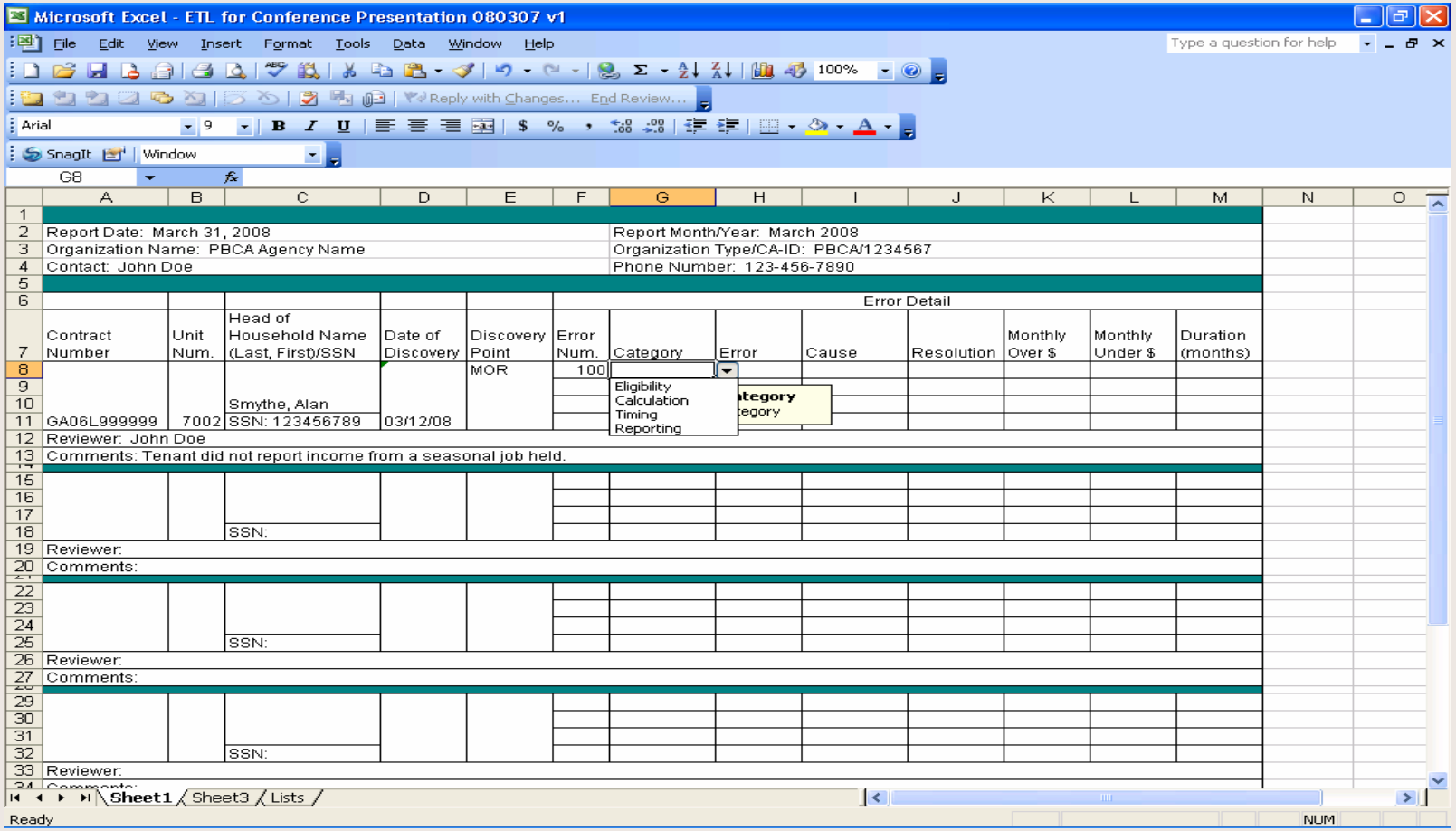
The most commonly applied resolutions for errors discovered are:

- No Payment Error ($0)
- Repayment Made to Tenant
- Payment Recovered in full by HUD (Include $)
- Repayment Agreement to HUD (Include $)
- Payment Suspended Temporarily
- Payment Suspended Permanently
- Fraud – Repaid
- Fraud – Repayment Agreement
- Fraud (Fully Document and Refer to HUD OIG)
- Payment Unrecoverable
- Other

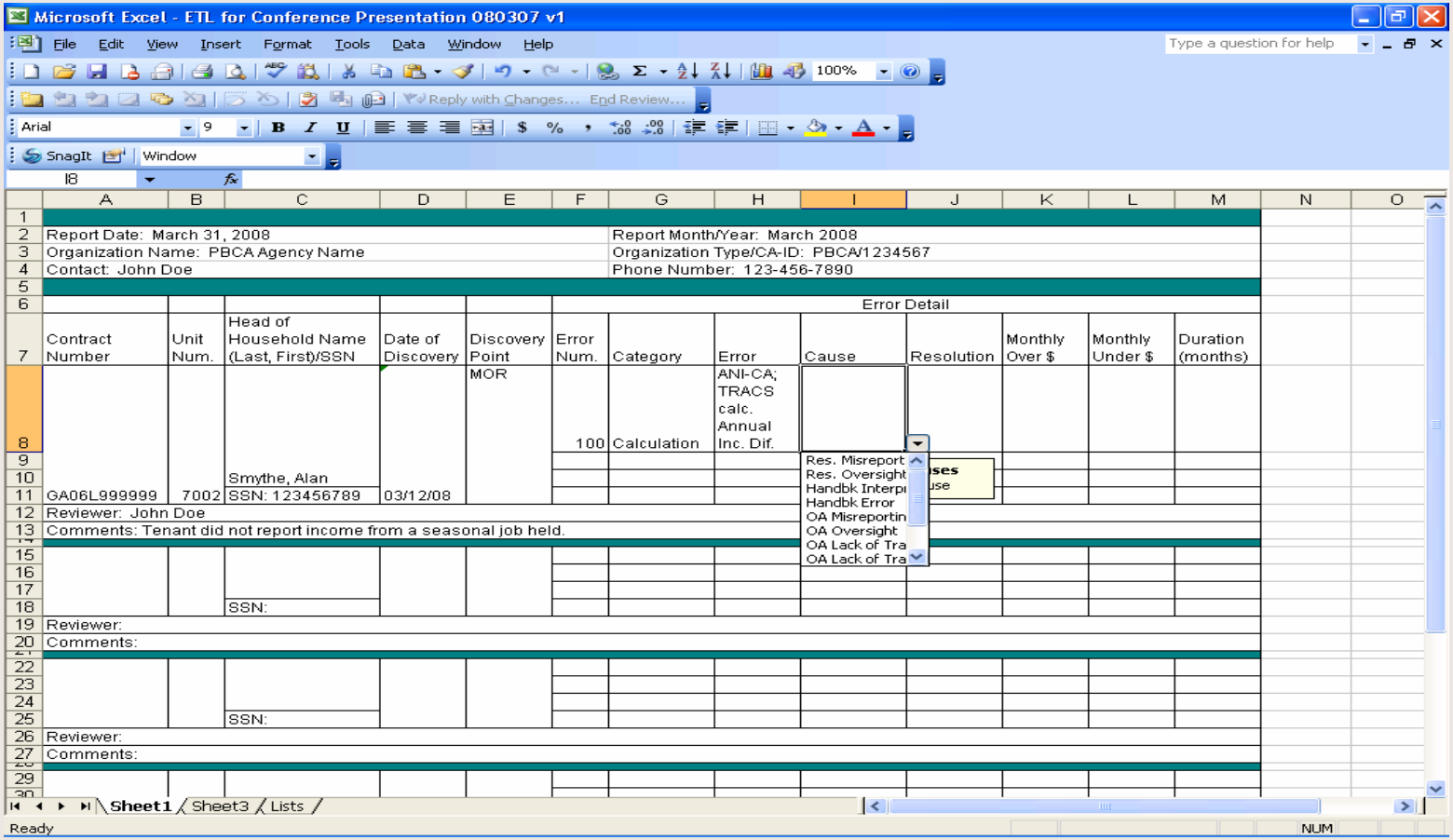
Where We Are Today Based on Working Group Progress



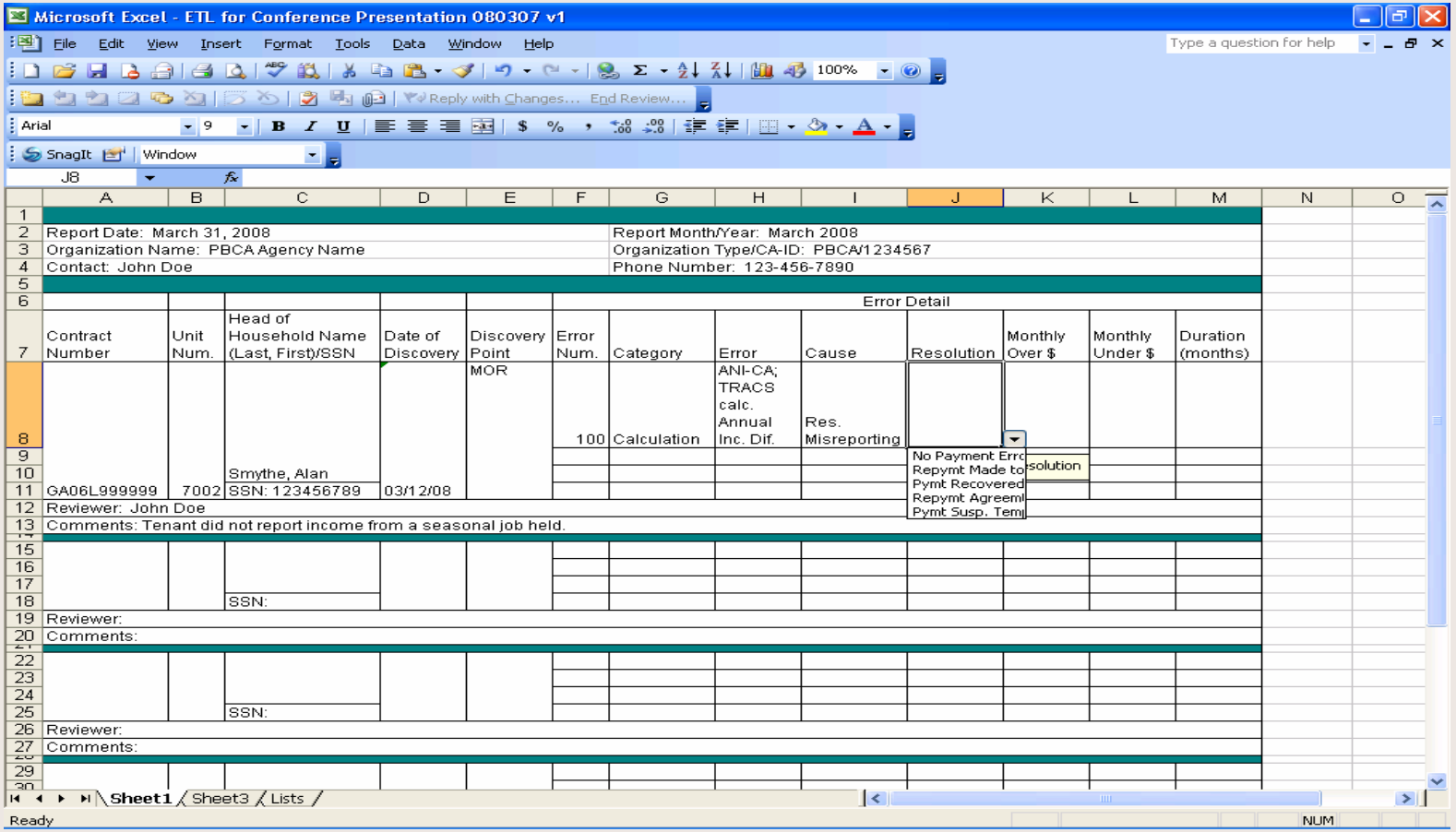
Where We Are Today Based on Working Group Progress



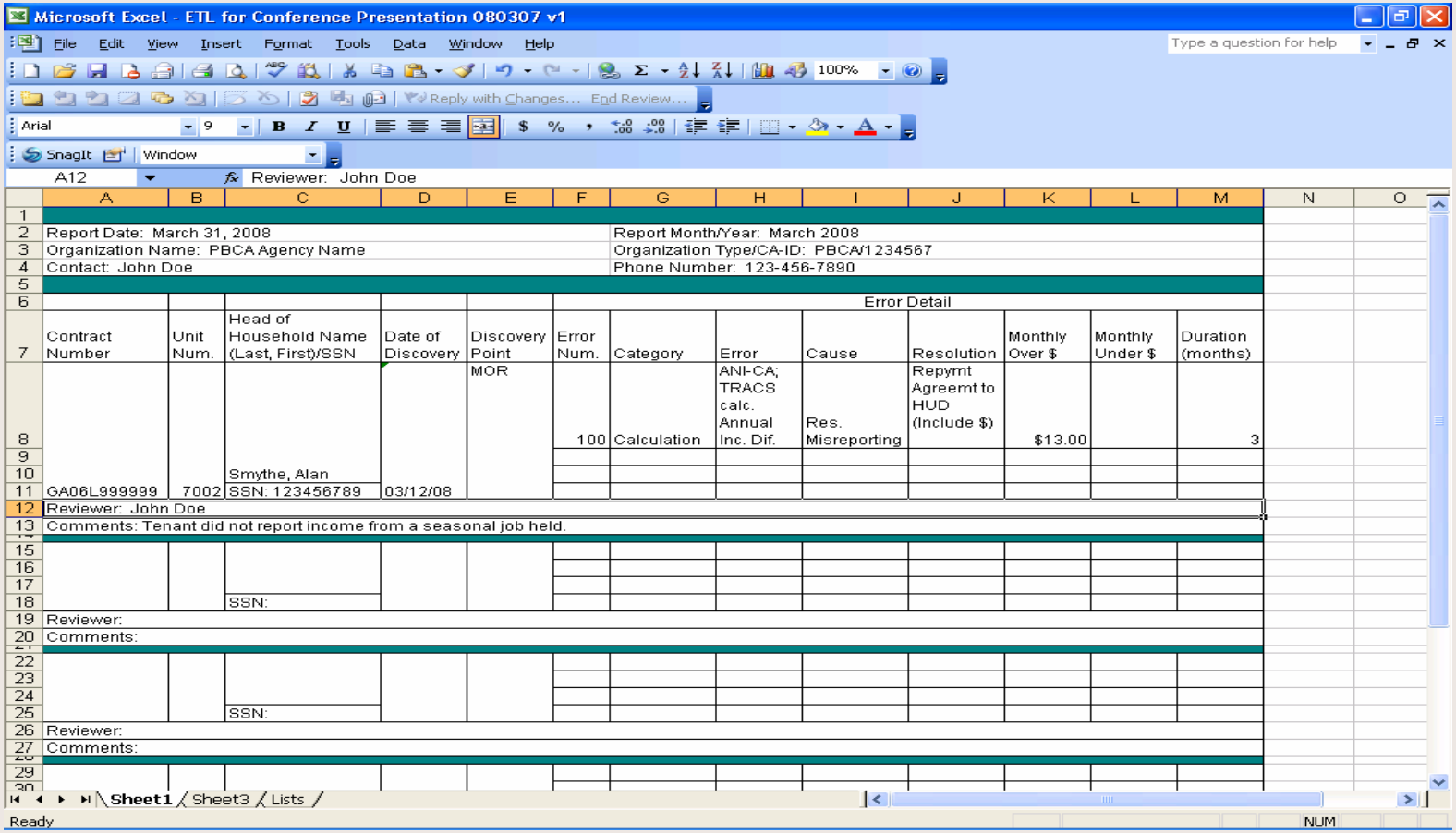
Where We Are Today Based on Working Group Progress



Where We Are Today Based on Working Group Progress



Where We Are Today Based on Working Group Progress



To-Be Process Model

The integration of the following components with a specific subset of procedural steps defines the basis for the To-Be Model for the Error Tracking Log (ETL). Organizations impacted by the recommended Model include: CA Software Vendors; CAs; ETL Processor Development Team.

- Tenant and Voucher, MOR and Resident Contact Error Data Feeds from CA
- ETL Processor and Centralized Database to Store Error Data
- ETL Query/Reporting Tool
- ETL Data Entry Tool for HUD/CAs

To-Be Process Model – Certification/Voucher Review

Tenant Certifications (HUD 50059) and Voucher Payment Requests (HUD 52670) are submitted electronically via TRACSMail.

- During this electronic submission, software applications used by contract administrators have internal data validation and error checks, in addition to some of those identified by TRACS. Error messages are generated informing the analyst of the problem.

- Analysts will work with the OA to take the appropriate action to correct the issue and resubmit the voucher data.

To-Be Process Model – Certification/Voucher Review

The To-Be Model will provide for:

- Transmitting error data that must be tracked/reported to the ETL, as they are identified by the software
- Transmitting error correction data to the ETL to be logged with the appropriate error, when a correction is submitted
- Parsing out and storing ETL errors on the database
- Pairing corrections with the appropriate original error as they are submitted
- Assigning status to each error tracked/reported.

To-Be Process Model – Certification/Voucher Review

Data received by the processor that can be paired with an original error will be classified as a “closed/resolved” error; any un-paired record will be classified as an “open” error. Precise processing rules for error types and pairing, statuses, and all other data elements will be defined during the development phase of the ETL automation.

To-Be Process Model – MOR Process

This process is an area where concrete, precise errors made at the OA or tenant level can be discovered. It is here that analysts must review all documentation completed and submitted by tenants (and management companies to CAs) substantiating tenant eligibility, income and rent, and subsidy amounts to be paid by HUD.

- MOR analysts must record each ETL-specific error discovered during the tenant file review as well as the additional data elements required by ETL.
- Errors will be logged only after they are resolved. Once the standardized list of errors and categories, along with all other data elements have been finalized and ETL data collection has begun, the CA must require their MOR analysts to log the detail that will be subsequently transmitted to ETL.

To-Be Process Model – MOR Process

The To-Be Model will provide for:

- Reporting of MOR errors and error details into CA software
- Transmission of error data to the ETL Processor via iMAX
- Parsing out and storing ETL errors on the database
- Assigning status to each error tracked/reported.

To-Be Process Model – MOR Process

PBCAs/TCAs will be required to record the appropriate information in a pre-defined MAT format to be submitted to HUD during monthly reporting of ETL errors identified during the MORs for the reporting month.

To-Be Process Model – Resident Contact Process

This process involves cases where residents may call their appropriate CA or HUD administrator to inform them of concerns that their rent may be incorrect, or that another resident may have misreported information impacting subsidy payment.

While it may or may not be a common occurrence, it has been identified as a possible source of discovery.

To-Be Process Model – Error Reporting

Recording of the ETL errors discovered during the certification/voucher review process will be automated through CA software. By implementing the ETL To-Be Model of a centralized database to capture and track required error detail, HUD will be able to run pre-determined formatted and ad hoc reports based on several criteria.

Similarly, MOR and Resident Contact error data from CAs using upgraded CA Software will be made available to HUD once it is entered.

To-Be Process Model – Error Reporting

Examples of Reports may include selection criteria such as:

- CA-ID
- Contract
- Project
- Error Type
- Error Category
- Range of Financial Impact – by Over-payment and/or Under-payment
- Range of Dates, etc.

Next Steps to Achieve the To-Be Process Model – HOW?

Restated, several components are necessary to fully satisfy the requirement of storing error detail in a centralized repository that can be used for HUD management to perform analysis and statistical reporting:

- Definition and Standardization of Error Data
- Centralized Database to Store Error Data
- MOR, Tenant and Voucher Error Data Feeds from CA and CA Service Bureaus
- ETL Processor
- ETL Query/Reporting Tool
- ETL Data Entry Tool for HUD/CAs

Next Steps - ***Task 1 [Define ETL Errors and Error Detail]***

The definition of errors and elements of data required for tracking must be completed before any of the other tasks can begin:

- Finalize Definition of Errors and Error Data Elements
- Finalize Transmission Format of Data Elements
- Finalize Reporting Format

Next Steps - ***Task 1 [Define ETL Errors and Error Detail]***

A precise, standardized definition of the following items must be agreed upon during the execution of this task to ensure that all data is consistent across CAs:

- List of errors and which categories they fall into (note: voucher adjustments will not be included as part of the Certification/Voucher error detection)
- Definition of what constitutes an error (i.e., length of time it existed, cap on dollar amount, etc.)
- List of contributing factors
- List of applicable resolutions

Next Steps - *Task 2 [Software Vendors Self-Assessment]*

Upon completion of Task 1, CA software vendors begin assessing the levels of effort to modify their software. Sources of data required from CA software include:

- Certification/Voucher errors identified by their applications. TRACS errors identified by software will be transmitted to the ETL Processor via iMAX.
- Certification/Voucher corrections. All corrections to errors will be transmitted to the ETL Processor via iMAX.
- MOR Error Data. CA software vendors must determine how they will capture the errors, resolutions, and other error data for subsequent transmittal to ETL.
- Resident Contact Error Data. CA software vendors must determine how they will capture the errors, resolutions, and other error data for subsequent transmittal to ETL.

The ETL Processor task will define the specifications for the MAT format of the data files and the transmittal mechanism. Therefore, this task will not only be dependent upon the effort previously identified in Task 1, but also in Task 5.

Next Steps - ***Task 3 [Schedule CA Software Effort]***

When CA software vendors have identified the scope and levels of effort to provide data to the ETL Database, HUD and partners will agree to a project implementation schedule for the development phase of the automation.

Next Steps - ***Task 4 [Design ETL Database]***

When all error data requirements have been identified and finalized, the task of designing the ETL database may begin.

Items to consider during the execution of this task include:

- Identifying and storing all data elements
- Identifying the best method of data retrieval (i.e., real time retrieval or scheduled data feeds from CA software)
- Data Volume

Next Steps - ***Task 4 [Design ETL Database]***

Additional considerations:

- Performance
- Possibility of Future Interfaces (if any)
- Common, integrated relational database and reference tables for all relevant data

Next Steps - ***Task 5 [Design ETL Processor]***

The ETL Processor will not make the determination that an error is an error, rather it will simply log information received. This Processor will take in all error data from CA software, and using a parsing algorithm (to be defined) retain only those errors identified by HUD.

As analysis of the ETL errors progresses, it may be determined that the subset of errors should be refined. This Processor will be designed to enable refinement of the standardized errors to occur with minimal impact to CA software in order to have one entry point of ETL error definition.

Next Steps - ***Task 5 [Design ETL Processor]***

Requirements of the ETL Processor Task:

- Parsing out ETL-specific errors from CA software errors transmitted
- Pairing resolutions with errors
- Flexibility in error parsing algorithm
- Identifying transmittal mechanism
- Identifying format of data files
- Identifying and documenting finalized set of business rules on pairing corrections with errors; resolutions; statuses, etc.
- Informing sender of status of transmission (successful, unsuccessful, invalid data)
- Validating all requirements

Next Steps - ***Task 6 [Design ETL Query/Reporting Tool]***

A Web-Based Error Tracking Tool can be designed for use by HUD Management for monitoring, extracting data, analysis and trending capabilities, as well as retaining historical data. A complete set of business rules required for pre-formatted and ad hoc reporting must be defined. Examples of data selection criteria for reporting include:

- Error Category
- Error Type
- Status (open, resolved, in work, etc.)
- Over-payment
- Under-payment
- CA/CA Service Bureau
- Contract

Next Steps - ***Task 6 [Design ETL Query/Reporting Tool]***

This task must take into consideration the following:

- Standardized User Interface
- Secure Web-based access governed by HUD-defined roles, responsibilities and business rules
- Server and user workstations must comply with HUD standards
- Communications will be performed through the Internet (e-mail and Web Browser)
- Regular server backups will aid in recovering any data or system software in the event of a system failure

Next Steps - ***Task 6 [Design ETL Query/Reporting Tool]***

Additional considerations:

- Authorized users can print reports and ETL information to connected printers
- Authorized users can download reports and error information to a PDF file, Excel file, or view as an HTML page
- User Support
- Training

Future planning will include a data entry function for HUD administrators to enter MOR error data into the ETL Database, similar to the special claims application used by HUD staff to review and approve special claims.

Next Steps - ***Task 7 [Design ETL Environment]***

In order to efficiently define the ETL environment, the following must be taken into consideration:

- Size
- Scalability
- Cost
- Hosting

DDL recommends that this Long-Term Solution be deployed in two phases: First as a pilot, and after a successful trial period and HUD acceptance, the solution should be finalized and made permanent.

Summary of Benefits

Some of the improvements to existing capabilities that will be realized by HUD management are:

- An ability to identify potential improper subsidy payments indicating pseudo monetary savings, realizing the importance and significance of the CA roles during the administration of Project-Based Section 8 Contracts
- An enhancement to the level of CA performance monitoring
- An ability to identify administering agencies that may not be as thorough performing MORs
- An improvement to monitor and track CA efforts in their timely and effective resolution and reduction of errors