# Data Validation: A Brief Guide for the New Validator
## (With Perhaps Some Ideas for the Experienced Validator)

If you've heard of Unemployment Insurance Data Validation, it was probably described in terms akin to Winston Churchill's famous remark about Communist Russia, "a riddle wrapped in mystery inside an enigma." In short, you probably heard that it might be something valuable, but who can understand it, much less do it? This is a modest attempt to lift the veil of mystery that shrouds DV, to give the state validator a layman's peek at the works inside the black box that is DV, and DV's importance in helping ensure accurate UI data.

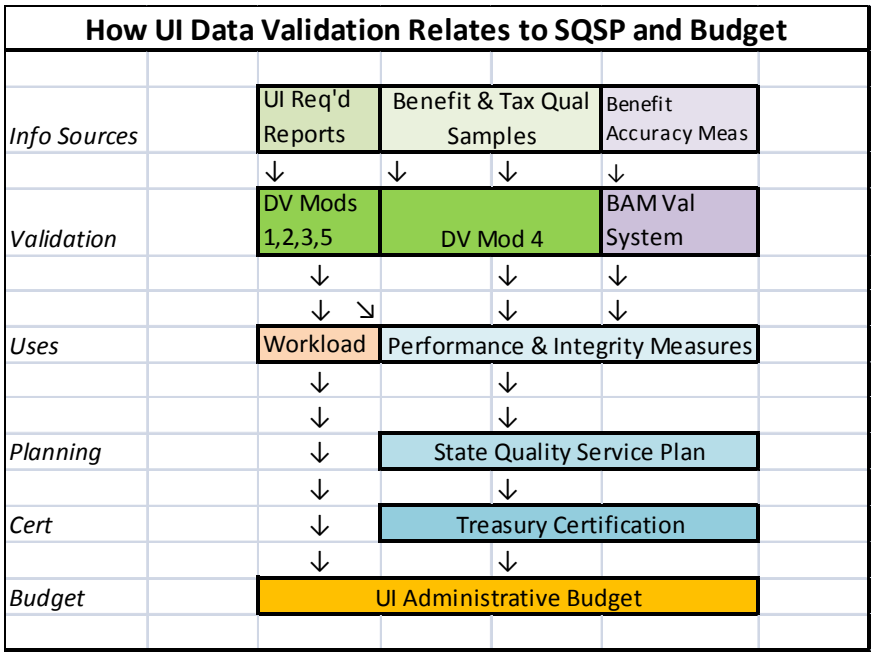
### The Concept, Structure and Development of Data Validation

Why Validate UI Data? The basic rationale for DV is pretty straightforward. Each state submits over 40 reports to the Department of Labor at intervals ranging from weekly to yearly. They encompass close to 3,000 different elements. Most of the reported data elements are simple counts, such as, State A reported taking 15,500 new intrastate initial claims last month. *It's not obvious from the number itself whether the true count is really 15,500 or not.* If important decisions ride on that number, it's crucial that State A really is taking the number of claims it reports. The same is true of the other states. Many of these reported elements *are* used for important purposes related to governmental or Departmental oversight, such as measuring performance, or setting and allocating the administrative budget, or serving as economic indicators. The Department knows it needs to be able to trust the numbers, and it's not alone. State administrators and all other users—from Fed Chairman Ben Bernanke on down--need to be able to trust what states report about their activities. With this fact in mind, the Department's Office of Inspector General and the Government Accountability Office insist that the Department be able to establish the validity of the key numbers it uses. How can this be done? How can we know whether that 15,500 count is right?

DV has a solution: build a separate record for each reportable claim the agency took in the month or quarter; sort them into the report categories; add them up; and compare that total with what was reported. If those records are built correctly, the independently "reconstructed" count will be the right one, and can be used to judge the correctness of the reported count.

The Unemployment Insurance Service builds the DV solution into its performance management system, UI Performs, and uses it to help ensure the accuracy of data used for budgeting and other key purposes. The following graphic helps to provide some context. It shows that the UI system relies on three main data sources for its conclusions about UI activities and their effects: UI Required Reports; Benefits and Tax quality samples; and the Benefit Accuracy Measurement (BAM) program. DV is designed to serve as the assessment tool for the UI reports data and the benefits and tax quality samples. (BAM has its own internal validation mechanism.) DV initially tells the user whether those data sources are accurate. In the case of reported counts it does so by providing an independently reconstructed count of what should be reported; for the quality samples it

indicates whether they were of the right size and randomly drawn from the correct universe; and for Wage Items (a tax workload count) whether UI wage records have been properly processed for counting. If the initial assessment shows inaccuracies, DV is part of a corrective process by pointing out where inaccuracies are occurring. Once they pass validation, these data sources can be used with confidence in the budgeting process and for other uses informed by UI data. This little guide addresses only the UI Required Reports validation segment of DV; it's the biggest and by far the most complex aspect.




How Records are Built. Every correctly-reported transaction has certain defining characteristics. DV's premise is to identify each characteristic and structure a record that contains a data field for each one, allowing someone to tell whether the record is reportable and properly classified by examining its characteristics. The agency must assemble that information for each type of record to be validated. For example, build a record so that someone can tell whether it has all the proper characteristics of a "new," "UI" "intrastate" "initial" claim, or instead is something else, such as an additional claim or a transitional claim that needs to be reported somewhere else on the same report or on another report. The sum of the records with all the right characteristics for "new UI intrastate initial claims"--the reconstructed count (or in DV terms the "validation count")--is what the state should have reported on line 101, column 2 of the ETA 5159 report. That validation count represents the standard against which the actual reported count of 15,500 can be judged. Those records serve as an "audit trail;" each one can be examined to ensure that the characteristics of the record correspond to appropriate agency documentation.

The Concept of a Population. When DV was designed, sixteen benefits reports and one tax report were selected for validation because they contain the information most relied upon for UI oversight, program administration and performance management. Within those reports, 334 key report elements or report cells were identified as key items

to validate. Examined by the type of transaction or status they represented, however, each of the 334 key elements was one of only 20 mutually-exclusive, non-overlapping types. DV called each of these types a "Population." In 2007, the department eliminated one of the benefits reports (ETA 9053); in 2012, it revised the ETA 227 report. As a result of these changes, the number of report cells validated is almost at 400 and 21 populations are used for the task. Table 1 below shows the relationships between Populations, reports and report elements validated.

DV approaches the validation of reported counts from the standpoint of the Population to which the reported count belongs, to take advantage of the 21-to-399 efficiency. The DV Population approach allows the validator to concentrate on one type of transaction at a time, and focus on a limited number of classifying data elements to make sure each record is properly built using those elements. On the other hand, many UI reports combine different types of transactions or status counts. As a result, UI validated reports and populations don't usually line up one-to-one, as Table 1 shows. For example, both the Benefits ETA 5159 (Claims and Payment Activities) and the Tax ETA 581 report (Contribution Operations) have five different types of key validated elements. Thus, building Benefits Population 1 (Weeks Claimed) validates only part of the ETA 5159. Validating all key elements on that report requires the construction of five Populations.

In designing each Population, every report element that the population would validate was carefully examined to identify the essential characteristics it must have to be properly reported. For example, Table 1 shows that nine of the cells on the 5159 report that we want to validate are counts of Weeks Claimed, and thus belong in Population 1. In the design phase of DV we made sure that the data "record" includes data fields (a) to establish whether the transaction can be traced to a known individual or business; (b) to establish whether it is a reportable transaction; and (c) for each characteristic needed to properly classify each of those nine report counts to be validated. Table 1 shows the number of data fields *extracted from the state's database* each population record requires. To validate the 399 key report counts, DV requires the states to build records that may contain as few as five data element fields (Higher Authority Appeals Case Aging) to as many as 20 (Field Audits). (The actual record contains two additional elements, an observation or sequence number assigned when the extract file is built and an optional field for the validation team's use.)

The Subpopulation. Based on the values in the record's data fields, the software sorts the records within each population into unique subgroups called "subpopulations"—456 for DV as a whole. The subpopulations are the components or building blocks for the reconstructed "validation counts" that tell what the 399 reported counts *should be*. The relationship between the subpopulations and the validation counts varies. In some cases, the validation count for a report cell requires only one subpopulation; in others, several subpopulations must be aggregated to make up the validation count for a single cell. In many cases, a subpopulation is a component of validation counts of multiple report cells on more than one report. With fairly minor expansion, this DV scheme could be modified to expand the number of validated reported cells to over 1,400 by validating individual time lapse counts. (DV concentrates on validating the totals; examination of state

reporting systems shows that if totals are reported correctly, time lapse reporting is rarely wrong.)

| Capsule Overview of the Scope of UI Data Validation | | | | | |
|---|---|---|---|---|---|
| Population | | Database Elements in Extract Record | Number of Subpops | What's Validated | |
| Number | Type of Transaction/Status | | | Number of Rpt Cells | On These ETA Reports |
| ***BENEFITS*** | | | | | |
| ***1*** | Weeks Claimed | 9 | 9 | 9 | 5159 |
| ***2*** | Final Payments | 10 | 4 | 13 | 5159, 218 |
| ***3*** | Initial Claims & Monetary Determinations | 12 | 46 | 39 | 5159, 218, 586 |
| **3a** | Additional Claims | 10 | 6 | 6 | 5159 |
| ***4*** | Payments | 16 | 51 | 48 | 5159, 9050, 9050p, 9051, 9051p, 586 |
| ***5*** | Nonmonetary Determinations | 12 | 70 | 64 | 207, 9052 |
| ***6*** | Appeals Filed, Lower | 6 | 2 | 2 | 5130 |
| ***7*** | Appeals Filed, Higher | 6 | 2 | 2 | 5130 |
| ***8*** | Appeals Decided, Lower | 14 | 55 | 19 | 5130, 9054L |
| ***9*** | Appeals Decided, Higher | 13 | 23 | 12 | 5130, 9054H |
| ***10*** | Appeals Case Aging, Lower | 6 | 7 | 1 | 9055L |
| ***11*** | Appeals Case Aging, Higher | 5 | 6 | 1 | 9055H |
| ***12*** | Overpayments Established by Cause | 13 | 27 | 30 | 227, Section A |
| ***13*** | Overpayments Reconciliation | 9 | 57 | 57 | 227, Section C |
| ***14*** | Age of Overpayments | 9 | 24 | 24 | 227, Sections C,E |
| ***15*** | Overpayments Established by Detection Method | 7 | 21 | 36 | 227, Section B |
| ***Totals*** | | ***157*** | ***410*** | ***363*** | ***15 Reports*** |
| ***TAX*** | | | | | |
| ***1*** | Active Employers | 16 | 2 | 2 | 581 |
| ***2*** | Report Filing | 10 | 16 | 6 | 581 |
| ***3*** | Status Determinations | 13 | 8 | 7 | 581 |
| ***4*** | Accounts Receivable | 13 | 16 | 10 | 581 |
| ***5*** | Field Audits | 20 | 4 | 11 | 581 |
| ***Totals*** | | ***72*** | ***46*** | ***36*** | ***1 Report*** |

Within each population, the logical flow is like this:

Each **Population** comprises **Individual Records**
**Individual Records** are placed into **Subpopulations**
**Subpopulations** are combined into **Validation Counts**

Once the validation counts are assembled, the reported counts are compared with them in the "**Report Validation**" phase, as follows:

### Validation Counts ↔ Reported Counts

The DV software retrieves the reported counts from the UI Database to save validators the effort—and risk of inaccuracy—of data entry. If the reported counts are within the selected tolerance limits of the validation counts, the reported count is considered to be valid. These tolerance limits are ± 2%, except for reported counts used in Government Performance and Results Act (GPRA) indicators; their tolerance is ± 1%.

## The Process View of DV

The DV journey begins with a tour book and a map. The tour book is the Generic DV Handbook. There's one for Benefits and another one for Tax because they're like somewhat different countries. Take the tour. You'll undoubtedly find the Handbook intimidating on a first view. There's no denying it: DV is a complex process, and the handbook cannot help but reflect that complexity. However, as with any complex system, the key is to get an overview of the basic flow of the process, and then to break it down into its component sub-systems and understand the reasoning behind them. That's the purpose of this brief guide.

In capsule form, the process for validating reported counts involves four basic steps:

1. Build an extract file
2. Test the extract file
3. Use the tested extract file to assess reported counts
4. If the reported counts do not match the DV standard, use DV results as a guide to fixing the process by which reported counts are compiled.

### *Step 1: Building the Extract File*.

The Handbooks both tell you that the first state product in validating a population is the development of the "extract file." That's DV's term for the set of those records mentioned above for every individual transaction you want the software to count up. It's produced by pulling out or extracting the necessary data from the state's UI database or management information system.

The programmer needs two things to build an extract file. The first in use the Record Layout, which tells which data elements the extracted record must contain. Record Layouts are in Appendix A and B of the DV Operations Guide (they're also available on the DV Web site at www.ows.doleta.gov/dv, and off the Population link on the first validation screen of the DV software.) As noted above, when you look at the Benefits Record Layouts in Appendix A or B you will see that for each Population the number of data elements is two greater than what Table 1 indicates. That's because these two elements are not extracted data elements: one is an observation number, which the programmer assigns when he or she builds the file; the other is a "user field" you fill, or leave blank, as you see fit. A closer look reveals that a few elements can actually be filled by the DV software. So, in rough terms, to build the 16 Benefits populations, about 150 elements must be extracted. Sixteen of this number is one element that appears 16

times: the Social Security Number (SSN), which every benefits record contains. Every Tax population record includes the Employer Account Number (EAN). Many of the elements repeat on each population such as Program Type and Intrastate or Interstate for Benefits, and Employer Type for Tax.

The second component is guidance on where those elements actually exist in the state system. DV has you build just such a guide; it's a ***map*** called **Module 3** of the DV Handbook. Module 3 gives the definitions for each of those 150 or so DV elements, and when completed tells where to find them in your state management information system or database. Actually, it may be better described as a combination of a ***map*** and the ***template*** for a map. We say "template" because part of the validation task may be to find the missing or current element in your state system that corresponds to the rules and definitions in Module 3. About 10 or15 years ago, Mathematica Policy Research staff met with every state's programmers and program specialists and actually identified each one of those items—if the state system had it, that is--and completed the first Module 3 mapping for each state. By the time they left town, your individual Module 3 map was as complete as it could be at that time: what that element was called in your state database and on what screens in your system you could find it. By now many things have probably changed.

*Updating Module 3*. Thus, ***ensuring that Module 3 is up to date is your first step in undertaking DV***. Module 3 is now maintained as a Web-based database application on a Department of Labor server. On the DV web page (http://www.ows.doleta.gov/dv/) you will find instructions and downloadable materials for establishing an account on that server. Once you have access to Module 3, you will find the Federal template and when you bring up your state's version you will see the most recent information available at the time the transition was made from the Access-based system operated by the DOL DV team to the Web-based system. The Web page also has a link to a tutorial for using the current version, showing you how to edit what is there and create a .PDF version for your PC or desk.

Updating a Module 3 is an opportunity for cross-program bonding, because it *requires a team effort*: database specialists, program specialists, programmers and other colleagues may be required to get it right. We hear that this often brings together many folks whose paths rarely, if ever, cross. Together you must begin working through it, element by element. Pick their brains and mine their institutional knowledge to update the data names, business rules, and locations of data elements. Then update the Web-based version. You are required to review Module 3 annually and certify that it is up-to-date. (The software cleverly provides a certification box that appears on April 1 and disappears on June 10 when the certification window for the year closes.) Once updated, the map is ready to guide you.

*Selecting the Population*. The next step on your journey is selecting the first destination. In terms of *importance*, Populations 4 (Payments) and 12 (Overpayments Established) of Benefits and Population 3 of Tax (Status Determinations) are highest because those validate the elements used for Government Employment and Results Act (GPRA)

indicators. However, they are not the easiest ones to get right and so another population may be a better starting place. Let's say you choose Population 1, Weeks Claimed. We validate the counts of Weeks Claimed reported on the monthly 5159 report; the handbook says you'll need a month's worth of transactions. With Module 3 in one hand and a Population 1 record layout statement in the other, head over to your IT shop to find a programmer. With any luck, it will be one of the programmers involved in revising your Module 3, and who's already familiar with it. Explain that you want to validate Population 1 (weeks claimed), for the month of June 2012. He or she is to build you a file of every week claimed transaction with Date Week Claimed between June 1 and June 30, 2012. Each record in that file will contain 11 elements. Nine must come from your state database, and eight of those elements must be filled--not blank--in every record (the layout says "required" and "not null"); the others are optional. The record layout gives the programmer the key information either in the table or in the header. Make sure he reads it all, including the part about the *secondary codes*—the part about the "dash and the state-specific value." The layout gives the Module 3 reference, telling him where to find each of those elements for the extract file.

The record layouts and Module 3 give the basic guidance for the programming phase. However, most programmers will also want the guidance of knowing what the DV software will be doing with the records. That is explained in Appendix A of the generic handbook. Appendix A defines every subpopulation into which the software will put records based on the values contained in the record's data elements. (Appendix A is essential for diagnosing why the software refuses to accept certain records. See below.) Some populations also have nuances that are explained in Appendix A notes.

*Loading the Extract File into the Software*. A couple weeks later the programmer sends you an e-mail with a humongous text file attachment. Here's your Population 1 file, Mr. Validator. Out of curiosity, you open it in Notepad. It contains 240,000 records, big strings of numbers, letters, and partial words separated by commas. It's the next best thing to gibberish. How to start making sense of it?

The most straightforward way is to use the DV software on your state Sun computer. If you don't have access to the DV software, contact your Sun system administrator or liaison to get access. Remember the name: you and he or she may have more than one contact during the DV process. Read the *DV Operations Guide*, available for download from the DV Web page. The *Operations Guide* will assume that you have given the file a name and asked the administrator to put it into the ***/opt/dv/data*** directory on the Sun machine--that's where DV files must reside--and that you have gone through the process your state has established to get access to the Sun computer and from there to connect to the DV software. You will get a User Name and a password. The Operations Guide will step you through the process of logging in with your login name and password and how to load the file. If the file is built according to the specifications, you'll see a rolling count of the number of rows loaded and errors as the load proceeds. Your file will probably take about 10 minutes to load.

Or maybe not: the file might not load. In that case, you'll be on the phone or e-mail within minutes to ask your programmer why the file did not load. He can probably help interpret the message that you got with the load failure. If not, contact the National Office ***Hotline*** at ***1-800-473-0188*** or the DV team by e-mail at dvrpts@uis.doleta.gov. They might ask for a sample of your records to help diagnose the problem.

Assuming the file does load smoothly, or that you've worked out any glitches that kept it from loading, now you have the file in a place where it's manageable. Although you aren't ready to take the results seriously, you'll first want to see the comparison between the validation counts (the software's independently reconstructed version of report counts, based on your extract file) and the actual reported counts—a sneak preview of ***Report Validation***. That will probably tell you whether or not you're in the right ballpark with the Population 1 file you've had built.

### *Step 2. Evaluating and Cleaning the Extract File*

Once a file is built and loads, you must evaluate or test it to determine whether the results from it can be trusted to represent what your state should be reporting of this transaction type. Without testing to determine that the validation counts are sums of the right things, the report validation phase would be just a comparison of two counts, both of which could be wrong. This is done in two steps: First, get it to where the DV software considers it properly built because every record fits into one of the population's subpopulation boxes. Second, ensure that all the elements in those records the software has accepted have values consistent with Federal reporting definitions.

*Step 2a. Dealing with the software's error cases.* The first step in evaluating the file is to look at the number and type of errors by clicking on the *View Errors* option on the software's Benefits Selection Criteria screen. The software presents up to 1,000 errors in total, in screens of 100. It rejects transactions as errors for three main reasons.

1. The first are *syntax errors*—some dates may not be formatted correctly, or there are misspellings in key field values, or crazy characters have crept into some of the fields. Error messages will point out syntax errors to you. In some instances, serious syntax errors can cause a population not to load, although the most common reason for a file not loading is that it does not contain the right number of data elements or "data fields."
2. The second are *assignment or "parsing"* errors. The software assigns transactions to its various subpopulations on the basis of the relationships among the elements in a record. These relationships are spelled out in great detail in Appendix A of the handbook, and identified in lesser detail on the "View Validation Counts" screen in the software. If the values of those elements are not in the expected relationship—key data are missing; dates are out of range; the relationship among elements is out of synch with the requirements—the record is rejected with a message saying it doesn't fulfill subpopulation criteria. That's often the hardest error message to interpret because it covers so many conditions.

3. Finally, there are *duplicate records*. These records have no syntax errors and they fit a subpopulation; unfortunately, they have identical twins or triplets. Appendices C and D of the *DV Operations Guide* give the criteria the software uses to determine duplicates. You have to examine duplicates and keep the one legitimate record, then rebuild the file without the "true" duplicates.

Examining the error file is an indispensable tool for identifying problems in an extract file, and finding key variables that might be missing or records with misspecified items.

> HINT: The software isn't your only tool for examining and assessing your files. A close second is a spreadsheet; you probably have Microsoft Excel. Say you're waiting for DV software access or are in the midst of conversations with programmers, help desks and hotlines. You can always look at part of the file yourself in Excel. Open the file in Notepad and highlight a reasonable sampling of records, say, 1,000. (Old versions of Excel accommodate only up to 65,000 records, but the 2007 and later versions can handle up to 1 million records). Copy and then paste the records into a fresh Excel worksheet. Use the *Data/Text to Columns* feature to "parse" the records into columns. Follow the prompts to parse a delimited file (i.e., one in which the data are separated by characters such as tabs or commas; DV files use commas) into the worksheet. Now, instead of the maze of numbers, letters and commas, all the data in the file are neatly arranged into columns so you can make some sense of them. You can insert a row at the top and use it to put the names for the elements. Take a look at the rows. Does everything follow the record layout? Excel allows you to sort by any column you want. Take a look at the dates: are all the dates in Field 7, Date Week Claimed, in the range you want? If something's amiss, work with the programmer to straighten it out.
>
> In our example of examining the errors, you can always select your errors from the software screen, copy and paste into Excel. They go in very neatly and there you can easily sort, or add comments, or do whatever you want. One big advantage of Excel is its flexibility in printing. Many browsers won't allow you to print your entire error (or subpopulation or "Source File") record, even in Landscape. The solution: Excel. Its print-to-fit capability is a godsend. If you're not familiar with Excel, take some training or have someone teach you. It's an essential tool in the validator's toolbox.

Your job in this step is to examine the errors and sort them into two groups: records that do not belong in the file because they are not reportable, and those that do but are for some reason incorrectly built. You want to eliminate the first type of errors and fix, and reinsert in the file, the second types. So, now let's assume that you've examined the file carefully using the software and your spreadsheet, and with the help of your programmer, you've made all the corrections you can think of. You've corrected or eliminated records that have syntax errors. You've isolated the duplicate records identified by the software, and removed the ones that appear to be true duplicates and reloaded the legitimate record

of the pair or multiple. The validation counts from the software are reasonably close to your reported counts and your programmer is confident that no transactions have been overlooked in building the file you're using. The next step is to look at what's really in those records that the software has pronounced itself satisfied with.

*Step 2b. Data Element Validation (DEV).* Data Element Validation involves digging deeper, testing and attesting that the file is really built properly so that you know whether you can say with some confidence that the counts from the file are based on records whose elements meet Federal reporting definitions. The DV methodology has formal methods for testing and attesting to the fact that an extract file is properly built. This is accomplished by reviewing a sample of records from each population. The term "attest" is used advisedly here, because the sample tests show *yourself and others* whether your files are built properly from data elements that conform to Federal reporting definitions. If they do, counts that the software produces from this file are the true standard against which to measure your reported counts. If not, more work lies ahead in building a file that contains only legitimately countable transactions.

whereas building and refining the extract file will involve mostly programmer time, DEV--especially Benefits DEV--is probably your most labor-intensive step as a validator. You do this by going back to original sources and—guided by Module 3--confirming that elements in the record come from the correct places and that those "places" are consistent with Federal report intentions. Although the Benefits and Tax methodologies do this somewhat differently, the purpose is the same: *to assure yourself and the Federal government that the extract file is clean and thus that totals computed from it can be trusted as the standard for judging whether reported counts are correct or not*.

- Benefits DEV relies on a series of samples, called "Random", "Missing subpopulation," "Minimum" and "Outlier." Their purpose is to examine the most significant elements used to build the extract file to ensure that the elements are properly selected from your MIS system or database. Some of the random samples are as large as 200 cases, although they are investigated in two stages so that if the records are very good or very bad you will know after reviewing only 60 cases. (The smaller samples are 100 in size, with a first stage of 30 cases.) The feds only require you to submit results of the random samples as your attestation; the other samples are for your own information, to give you insights into other parts of your population file that may have errors. You need to do them all, submitting random results and completing and saving the others—all as the means of checking to ensure that your extract file is built properly. *An RV result for a population is not considered valid until all that population's random samples have passed.* Starting in VY 2009, the RV and samples must come from the same extract file. Since DV software version 2.0, the DV software has enforced the requirement that both RV and random samples come from the same file by not allowing results to be transmitted until all random samples are completed.
- Tax DEV has some differences from Benefits. The principle of allowing only a tested and proven extract file for the derivation of RV results—i.e., requiring that both DEV and RV results must come from the same extract file—was first

established with Tax validation. In the DEV phase, Tax validation tests extract files for quality differently than Benefits:

- Whereas most benefits populations use large random samples, Tax DEV uses very small samples (called Minimum or File Integrity or "FIV" samples of only two records per subpopulation) to test whether the data elements come from the correct locations in the database.
- Tax DEV supplements FIV samples with a series of "range" tests to determine whether Federal primary codes used to build an extract file—such as "N" for New Status determination, or "S" for successor, etc.--are consistent with your own state's multiple codes. (Not all states have multiple codes; if not, this test doesn't apply as there is a 1-to-1 mapping from a state code to the DV letter code.) But many states have numerous identifying codes, or use ranges of Employer Account Numbers to identify "Contributory" or "Reimbursing" employers. DV uses queries and distributions to help you assess the integrity of your file by telling you whether the Federal codes are supported by all your state codes.

To pass a tax population, you must first pass all the diagnostics that tell you the file is built properly and then you must pass the RV results for *that same extract file*. Pass or fail, the software requires that all diagnostics be done before any results can even be submitted. Benefits now follows the Tax model.

***Step 3. Evaluating Reported Counts with a Tested Extract File***

But we digress. Back to Benefits Population 1. You've built your file, you've done your DEV diagnostics work and entered all the data into the software. If all your random samples pass, all your smaller "non-random" samples are clean, and your RV results are within ± 2% of your validation counts, you're done. You have passed validation for Population 1. Submit it by the June 10 due date and you don't have to validate it again for three years (submit it late and you'll have to repeat the exercise the following year). You have demonstrated in a reasonable way that what your state reports is accurate. But what do you do if things *don't* match? How can you tell whether a discrepancy lies with your validation efforts, or your reporting…or both?

***Step 4. Addressing Report Validation Discrepancies***

No question about it, this is where the task can get tricky. Your objective as a validator is not to get that "pass" trophy for your office wall for its own sake but to ensure that your state reports correctly. To do this, you have to be able to identify the reason for a discrepancy between validation counts and reported counts, and make recommendations for correcting reported counts if that is the problem. To make such a recommendation, you have to be able to demonstrate that the reporting system *is* the problem. To do so will require much thought and consultation with your colleagues, especially the folks who designed the reporting system (if they are still with the agency, that is; they may have departed years ago!) and your extract file programmer. But the following decision table may help guide your collective thinking.

| Table 2<br>Drawing Inferences from Different Random Sample and RV Results | | | |
|---|---|---|---|
| ***Random Sample*** | | ***Report Validation Computation*** | |
| ***Result*** | ***Inference*** | ***Result*** | ***Inference*** |
| Pass | Extract file built properly; database probably OK; but universe of transactions may be too small | Pass | Report Counts OK; or both report counts and extract file omit some transactions |
| | | Fail; reported counts < validation counts. | Definite problem with reported counts |
| | | Fail; reported counts > validation counts. | Report counts probably in error but extract may omit some transactions |
| Fail | Extract file bad, or Database bad, or Both are bad | Pass | Cannot conclude that report counts are valid |
| | | Fail | Report counts may be OK if sample failure indicates extract file incorrectly built |

- Case 1: Both random samples and RV are "pass." You can reasonably conclude that your reported counts are correct, although there is an outside chance that both validation and reported results are understated. The inherent weakness of the DV methodology is that you may miss some transactions when you build your extract file, and if they're not in the file, you can't count them or assess them. The chance is probably small, but it is something you need to be aware of.
- Case 2. Random samples pass but RV fails—reported counts are not within the 1% or 2% tolerance of validation counts. You would conclude that your extract file and database probably both accord with Federal reporting definitions, and that you probably have a reporting problem to fix. However, your certitude may vary:
    - If your reported counts are *less* than your validation counts, you certainly have a problem with the way your reported counts are generated.
        - Remember, if there is a problem with a "clean" extract file it is that it fails to include transactions.
    - If your reported counts are *more* than validation counts, you probably have a problem with the way your reported counts are generated, but you are less certain because there is always the outside chance that your programmer failed to include some transactions.
        - Before you conclude that your report-generating software is wrong, consult with your programmer to make sure that your extract file includes all transactions. Examine your error file; make sure that transactions the software rejected were rejected correctly and were not rejected because of minor issues with otherwise countable transactions.
- Case 3. Random Sample(s) fail. If your random sample(s) fail, you can't really draw any conclusions about your reported counts because you have no assurance that your standard—the extract file counts—is reliable. You'll have to determine whether the problem lies with your extract file—it's not picking up the correct data from the database—or with your database, or both.

- Case 4. Mixed Random Sample Results. Because some populations have multiple random samples, you could end up with a mixed case—some samples passing, others failing. In most populations, the pass-fail groups line up with random samples, so you can draw conclusions about the report cells validated by those groups with passing random samples, and concentrate your efforts on fixing the portions of the extract file, or portions of the database, where random samples do not pass. All random samples, and all RV groups, must pass before a population can pass, but you can segment your work within many populations.

**Light at the End of the Tunnel**

At some point you will conclude that you're at a stopping point. We all fondly wish you be at a "Case 1" situation in which your DEV random sample(s) and population RV pass. However, that may not be the case: you may conclude that your reporting software is faulty and you use your DV results to provide guidance to the report shop. Or, you may have done all you can with your long-suffering programmer to build your extract file and find that your database has deficiencies, and you've indicated those deficiencies to the appropriate office and asked them to put changes into the queue. In any case, hit the "Transmit" button to send in the DEV random sample if you haven't already done so, and transmit the RV results, so that your friends in the Regional Office and the National Office know the status of your efforts. If you've passed everything you're good for three years, unless it's one of those GPRA populations that must be done every year. In either case, win, lose, or draw, congratulations! You've done it. On to the next population. That wasn't so bad, now, was it?

**Wrapping Up**

Oh, if you have passed, just make sure you've done everything you need to wrap up and document your effort. Draw and examine the "nonrandom" samples; we trust they will confirm that everything is OK; if not, you'll have to look into what caused the problems they find. Make copies of your results—save screen shots of what you're sending in to the National Office via the software, save documentation of sample results, archive the extract file on which those passing results are based—and tuck them away where you can find them for ready retrieval in case of a regional Office review or some kind of audit. Now it really is on to the next one!

**Need Help?**

One more thing. This little tour assumed that you are the "go-it-alone" type who prefers to work from documentation. If you're of a different persuasion—and even if you're not--training is available. Mathematica Policy Research gave training to all, or nearly all, states in DV during the development period. But, that was long ago, and memories fade and the cast of characters changes. If you are in the position of starting anew, don't hesitate to request both retraining and ongoing assistance from the National Office DV team. We stand ready to provide whatever resources will allow.

**A Brief Checklist**

Review the following list of items to establish your DV marching orders for the upcoming validation year. The easiest way to see what is coming up is by examining the "Validations Due in VYxx" from the DV Web page.

| Data Validation Checklist<br>(Perform this review in July, as soon as "Validations Due" is posted to DV Web Page) | | |
|---|---|---|
| ***Item*** | ***Action Needed*** | ***Action Due by*** |
| Benefits Populations 1-3a; 5-11; 13-15 | Validate in current VY unless they passed validation 1 or 2 years ago; or if reports system changed within a year | Submit results by June 10 |
| Tax Populations 1, 2, 4, 5 | Validate in current VY unless they passed validation 1 or 2 years ago; or if reports system changed within a year | Submit results by June 10 |
| Benefits Populations 4, 12; Tax Pop 3 | Revalidate this year (GPRA Population) | Submit results by June 10 |
| Benefits and Tax Module 4 Quality Sample Validations | Validate in current VY unless they passed validation 1 or 2 years ago; or if reports system changed within a year | Submit results by June 10 |
| Module 5, Tax Wage Item Validation | Validate in current VY unless it passed validation 1 or 2 years ago; or if Tax system changed within a year | Submit results by June 10 |
| Benefits Module 3 | Review; update as needed throughout VY | Certify between April 1 and June 10 |
| Tax Module 3 | Review; update as needed throughout VY | Certify between April 1 and June 10 |