

The Instrument

The New York State Tool
for Administering
Training and Experience Exams
Online

The Basics

- Builder
- Collector
- Data Entry Collector
- Scorer
 - Administration

The Builder

- The **builder** creates a complex data structure from a well-formed collection of data described using XML. The XML data description specifies:
 - The data to be collected,
 - the captions used to describe the data on the web site,
 - labels to guide the web site visitor through the process,
 - validation routines to ensure the collected data is correct, and
 - scoring markers to describe how to evaluate the collected information.

The Collector

- The **collector** uses this complex structure to:
 - display the information on the Web,
 - collect the information from the visitor,
 - validate the information and request correction if necessary, and
 - optionally store the information in the database.

The Data Entry Collector

- The **data entry collector** uses the structure to:
 - display only scored questions on the Web,
 - capture the information from paper forms,
 - validate the information and request correction if necessary, and
 - optionally store the information in the database.
- If the data is stored in the database, then the **data entry collector** can also perform *verification*.

The Scorer

- The **scorer** evaluates the information entered by the visitor, applying the scoring rubrics and the banding table to arrive at a final score for the values entered.

Workflow (easy version)

- Candidate applies
- Local registers candidate for exam
- Candidate takes the exam
- State scores the exam
- State releases the exam when scoring is complete

Definitions

- *Instrument*: the entire on-line questionnaire, including all the formatting, validation and scoring parameters.
- *Section*: the information that will appear on one screen. An instrument is made up of one or more sections.
- *Question*: a single request for information. Sections are generally made up of related questions.

Definitions

- *Value*: what the candidate enters to answer a question. In sections that allow multiple answers to a question, *Value* is the sum of the answers, or the count of answers if the answers are non-numeric.
- *Response*: a single instance of the values provided by a visitor to answer all the questions in a section.
- *Points*: what we award a candidate based on the value.

Definitions

- *Multiple Section*: a section that allows the visitor to enter responses to its set of questions multiple times. Multiple sections are used to collect information such as classes or work experience that tends to have more than one instance.
- *Rating Scale*: the scoring rules for the instrument
- *Group*: a set of sections related by the rating scale. Groups are only used for scoring purposes.

Instrument Parameters

- Major Parameters
 - SECTION
 - GROUP
 - BAND

Instrument Parameters

- Display Parameters
 - RESETBUTTON
 - CLEARBUTTON
 - CANCELBUTTON

Instrument Parameters

- Wording override
 - CHECKLISTHEADLINE
 - ERRORHEADLINE
 - BUTTONSAVE
 - BUTTONCANCEL
 - BUTTONCONTINUE

Section Parameters

- Major Parameter
 - QUESTION

Section Parameters

- Behavior Parameters
 - MULTIPLE
 - FIRSTSECTION
 - DATAFLAG

Section Parameters

- Scoring Parameters
 - MINPOINTS
 - MININSTANCES
 - MINQUAL
 - MAXPOINTS
 - POINTS
 - MINVAL
 - GROUP

Section Parameters

- Display Parameters
 - RESETBUTTON
 - CLEARBUTTON
 - CANCELBUTTON

Section Parameters

- Wording Override
 - ERRORHEADLINE
 - BUTTONSAVE
 - BUTTONCANCEL
 - BUTTONCONTINUE

Group Parameters

- NAME
- MAXPOINTS
- POINTS
- MINVAL

Band Parameters

- RAWSCORE
- CONVERTEDSCORE

Question Parameters

- NAME
- TYPE
 - Label
 - Text
 - Daterange
 - Select
 - Comment

Question Parameters

- Captions
 - TEXT
 - SUMMARYTEXT
 - POSTFIELD

Question Parameters

- Validation
 - REQUIRED
 - REQUIREDIF
 - MESSAGE
 - VALIDATE
 - RANGE

Question Parameters

- Select options
 - CHOICES
 - ANSWER
 - ANSWERTEXT
 - VALUE

Question Parameters

- Display Parameters
 - DEFAULT
 - SIZE
 - MAXLENGTH
 - CLEAR
 - ROWS
 - COLS
 - BGCOLOR
 - BREAKBEFORE
 - BREAKAFTER

Question Parameters

- Scoring
 - MINQUAL
 - POINTS
 - POINTSPERINSTANCE
 - UNIT
 - ANSWERKEY
 - MINRESPONSE
 - MINVAL
 - MINPOINTS
 - BONUSPOINTS

Question Parameters

- Scoring
 - MININSTANCES
 - BASEPOINTS
 - BASERESPONSE
 - ALWAYSAWARDBASEPOINTS
 - MAXPOINTS

Question Parameters

- Scoring – Recency
 - RECENCY
 - RECENCYFIELD
 - CUTOFF
 - CURRENT
 - IGNORERECENCYVALUE
 - PARTIALPOINTS

Question Parameters

- Scoring – Concurrency
 - OVERLAP
 - OVERLAPDATE
 - FULLTIME

Introduction to Scoring

- *Scoring by Value*
 - If **Value** is 0:
 - **Score = Score + 0.**
 - If **Value** is less than **MinResponse** but greater than **zero** and **AlwaysAwardBasepoints** is **No**:
 - **Score = Score + 0.**
 - If **Value** is less than **MinResponse** but greater than **zero** and **AlwaysAwardBasepoints** is **Yes**:
 - **Score = Score + BasePoints**

Introduction to Scoring

- If **Value** is greater than **MinResponse**:
 - If **BasePoints** is greater than 0:
 - **Score** = (**BasePoints**) + ((**Value** – **MinResponse**) / **Unit**) times **Points**
 - Otherwise:
 - **Score** = (**Value** / **Unit**) times **Points**
- Since **BasePoints** is usually 0 and **Unit** is usually 1, the scoring formula most often used is:
 - **Score** = **Value** times **Points**

Scoring Sections

- Capping a section
 - MINPOINTS
 - MAXPOINTS

Example: a candidate needs to earn 5 points to get any value for a section, but the most we will give for a section is 2 points

MINPOINTS=5 MAXPOINTS=2

Scoring Sections

- Bonus points
 - MINVAL
 - POINTS

Example: Award 2 points if the total value of a section exceeds 4.

MINVAL=4

POINTS=2

Concurrency

- or -

Way too much detail on handling concurrent
employment experience

Concurrency

- Step One
 - Determine which experience activity would give the candidate the most points
 - Sort the potentially overlapping experience sections by **points**.

Concurrency

- Step two
 - Determine the range of dates covered by the work experience
 - If necessary, trim the “From Dates” to limit the experience to recent dates
 - **Recency** and **RecencyField** are used to trim the “From Dates”
 - If necessary, trim the “To Dates” to limit the experience by the cut-off date.

Concurrency

- Step two
 - Determine the range of dates covered by the work experience
 - Determine the earliest date of experience and the latest date of experience across the employment experiences listed
 - Subtract the earliest date from the latest date (and add 1) to determine the number of months covered by all experiences

- Step two

- Determine the range of dates covered by the work experience
 - Build an empty array, starting at the earliest date (array element 1), with one array element for each month across the range of experience

Concurrency

- Step 3
 - Credit experiences month by month
 - Start with the highest scoring experience
 - Calculate its starting month relative to the earliest month for all experiences
 - Calculate the ending month
 - Place up to **fulltime** hours in the array for the starting month, the ending month, and each month in between (we will call this “hours credited”)

Concurrency

- Step 3
 - Credit the experiences month by month
 - Do the same for each experience, with a twist
 - The twist is this: only place subsequent experience if there are hours "left."
 - For each month for which we have already credited hours, calculate the difference between **fulltime** and hours worked that month.

Concurrency

- Step 3
 - Credit experiences month by month
 - The twist
 - If **fulltime** hours have not yet been credited, then credit either the number of hours worked (for this next experience), or the difference between **fulltime** and hours previously credited, whichever is less

Concurrency

Job 1	Job 2	Job 3
15		10
15	20	10
15	20	10
	20	10
	20	10
		10

Concurrency

Job 1	Job 2	Job 3
15 (15)		
15 (15)		
15 (15)		

Concurrency

Job 1	Job 2	Job 3
15 (15)		
15 (15)	30 (20)	
15 (15)	30 (20)	
	30 (30)	
	30 (30)	

Concurrency

Job 1	Job 2	Job 3
15 (15)		10 (10)
15 (15)	30 (20)	10 (0)
15 (15)	30 (20)	10 (0)
	30 (30)	10 (5)
		10 (10)
		10 (10)

Concurrency

Job 1 (3 pts)	Job 2 (2 pts)	Job 3 (1 pt)
15 (15) 1.29		10 (10) 0.29
15 (15) 1.29	30 (20) 1.14	10 (0)
15 (15) 1.29	30 (20) 1.14	10 (0)
	30 (30) 1.71	10 (5) 0.15
		10 (10) 0.29
		10 (10) 0.29