

# Chapter 18

## Computer-based Platform

### INTRODUCTION

PISA 2018 maintained a computer-based platform as the primary mode of assessment of student skills. While a paper and pencil version of the assessment remained an option, development of new items according to the updated assessment frameworks was done only for the computer-based assessments. In the end, most countries chose to implement and deliver the entire survey on a computer-based platform. All domains were delivered via computer, including the optional financial literacy assessment. The computer-based assessment was delivered in a total of 114 different language versions across the participating countries/economies.

This chapter focuses on the functionality and technical implementation of the computer-based assessments. It also details the PISA student delivery system (SDS), integrated the assessments with the computer-based questionnaires for delivery of the PISA survey in schools. Finally, we conclude with a discussion of the open-ended coding system (OECS), used for coding of open responses.

### ITEM RENDERING

The items for PISA 2018 were implemented using web-based technologies: HTML, CSS and JavaScript®. Modern web browsers (such as the bundled Firefox v48) provide a rich set of functionalities for attractive presentations and engaging interactivity. At the beginning of the development work, an overall user interface was designed, with common elements such as navigation, help and progress indicators. The items were built in such a way that these common elements were shared, so that the same version was used in all items in each language version.

PISA 2018 items are generally grouped into units consisting of a set of items with a common stimulus. Each unit was constructed independently, with the questions and stimulus components developed first in English, then translated into French to create the two source language versions. The development was done by experienced web UI developers, using standard HTML components and adding custom functionality via JavaScript. Each unit could be viewed on its own or grouped with other units into a test form for delivery to students as part of the assessments.

In some cases, such as the interactive scientific literacy units, common functionalities were split out into shared programming libraries that could be reused in multiple units. For example, in the scientific literacy units, the experimental data tabling and management functionality was built as a shared library. The library also managed the recording of data and supported scoring of the student's performance based on unit specific criteria. Likewise, in reading literacy, the management and display of multiple source in tabs was encapsulated into a shared library.

The visual aspects of the PISA 2018 items and the automated coding of student responses were both implemented using JavaScript®. Shared libraries were created to implement this coding in a common way. The libraries targeted the various response modes used within PISA:

- Form: for all responses using common web form elements such as radio buttons, checkboxes, dropdown menus and textboxes.
- Drag and Drop: for items using drag and drop as the response mode.
- Selection: for items where the response is given by clicking on an object or region of the screen. This can be, for instance, clicking on part of an image, a cell in a table or a segment of text.
- Ad hoc: A general catch all that uses custom JavaScript<sup>®</sup> code to implement the coding. This was used for unique situations, such as coding for collaborative problem solving and interactive scientific literacy items.

In all cases except the ad hoc coding, the coding for a specific item was specified using rules composed of conditional expressions and Boolean operators. Each library implemented appropriate conditional expressions (e.g., a CONTAINS operator in the Drag and Drop library to test if a drop target held a particular drag element).

## TRANSLATION AND ONLINE ITEM REVIEW

Given the need to support over 110 different national language versions of each unit, automated support for integration of national translations and adaptations was critical. Supporting this process started when the units were initially developed. The HTML files that implement the display of the unit contain only HTML mark up and the text to be shown on the screen. Layout and formatting specifications are stored separately in CSS stylesheets. The text of the units is then extracted from these HTML files and saved to XLIFF (<http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html>) files. XLIFF is a standard file format used for computer supported translation. Once a translation is completed, the XLIFF file is injected into the original source version of the item, resulting in HTML files with the translated text of the unit.

One of the guiding principles of the platform was that the quality of a translation is improved if the translators can view their translation in the context of the items they are translating. In an ideal world, translators would work in a completely WYSIWYG (what-you-see-is-what-you-get) mode, so that they enter their translations directly in the items. But this is not technically feasible, and also may tempt translators to focus more on the visual aspects of the items, which are tightly controlled for comparability, rather than the translated text. A good compromise was to provide an easy to use preview capability, giving translators a quick way to view their translations as functioning items. The PISA 2018 Portal offered this capability. Users were able to upload an XLIFF file, either partially or completely translated, and in a matter of seconds they received a preview of the given unit in exactly the same view as a student would receive. From this preview, they could interact with the units in the same way as a student responding to the unit would. This was an important factor, particularly for the complex units of reading literacy and interactive science. This preview also allowed countries to test and identify potential problems with their translated units before receiving the final versions and software to be used in schools. Problems reported were fixed as early in the schedule as possible.

## SCHOOL COMPUTER REQUIREMENTS

The goal for PISA 2018 was, to the extent possible, to use the computers available in the sampled schools with no modifications. The PISA 2018 system supported Windows based computers in the field trial and both Windows and Macintosh computers for the main study. The following minimum technical requirements were established for both the field trial and main study:

- CPU Speed: 1500MHz
- Operating System: Windows XP, Vista, 7, 8 or 10
- Installed memory: Windows XP = 768 MB, other Windows versions = 1280 MB
- Available memory: 467 MB for Windows XP, other Windows versions = 774 MB
- Screen resolution: 1024 x 768 pixels
- USB transfer rate: 7.5MB/s.

These were the minimum requirements. Computers with higher capabilities would obviously perform better (e.g., respond faster) when delivering the survey, but the requirement listed above were the minimum settings that would provide adequate performance.

## SYSTEM DIAGNOSTIC

In order to verify that the available school computers met the minimum requirements, a system diagnostics application was provided to countries. The System Diagnostics is a version of the SDS without the tests and questionnaires. It is intended to be given to schools to check the compatibility of the school computers with the PISA software. It checks the computer's hardware and software setup and report this back to the user, typically the test administrator or technical support staff in the school. The system diagnostics was provided to countries approximately six months prior to the start of the field trial and main study. This allowed countries to provide the software to sampled schools to determine if their computers could run the PISA survey. Additionally, it was recommended that test administrators run the system diagnostics on the day of the test prior to conducting the assessment.

For cases where schools did not have adequate quality or quantity of computers, test administrators brought laptops into the school to augment the available infrastructure. In a few cases, countries chose to administer the PISA tests in all sampled schools on laptops brought into the schools. This avoided "surprises" on the day of the test, where computers were not available or not functioning properly.

## TEST DELIVERY SYSTEM

The PISA 2018 test delivery system, called the student delivery system or SDS, integrated the PISA computer-based assessments and questionnaires for a country, along with of a number of components packaged together to run as a standalone application on a USB drive. The SDS did not require network connectivity or external resources to operate. All software and data was on a single USB drive, and results were saved back to the USB drive. The SDS could also be run from the computer's hard drive if desired. The components which made up the SDS included the following:

- Apache web server (<https://httpd.apache.org/>)
- MySQL database engine (<https://www.mysql.com/>)
- PHP interpreter and libraries (<http://php.net/>)
- Firefox Portable web browser ([http://portableapps.com/apps/internet/firefox\\_portable](http://portableapps.com/apps/internet/firefox_portable)).

The actual test and questionnaire content were included together with these open source applications,. The PISA test delivery system was implemented to display this content to the students and collect the results, Using components of the open source TAO test delivery system (<http://www.taotesting.com/>) as a basis, the system was custom built for the needs of PISA 2018. This included implementation of the test flow, which assigns the designated test form

and questionnaires to a student, then sequences through the test units and questionnaires in the appropriate order. It also included the functionality for collecting the survey results and exporting them when the tests are completed. The PISA test delivery system was built primarily using PHP, with JavaScript used for interactive displays and communicating between the web browser and web server.

The system was launched by running a custom executable program (PISAMenu) written for controlling the delivery of the PISA tests. Separate programs were written for Windows and Macintosh. From this program, a test administrator could launch the PISA tests, launch the system diagnostics, or manage exported data files. These exported files are described below. Launching either the PISA tests or system diagnostics would start the web and database servers, then launch the Firefox web browser to begin the process. When the PISAMenu program was shut down, the various components of the SDS were also terminated.

The Firefox browser used for the PISA tests was configured to run in “kiosk mode”, so that it filled the full screen of the computer, making it difficult for users to access external applications when running the PISA test mode. A keyboard filter was also installed so that students could not easily leave or terminate the browser window, e.g., by pressing Alt-Tab, and switch to another program during the test. The keyboard filter did not completely block such attempts, though. It was also not possible to block the Ctrl-Alt-Delete sequence under Windows, as this required installation of a custom software driver at the system level. The goal was not to install any software on the school computers, so this driver was not used. It was expected that the test administrator would monitor the students during the test and watch for cases of students trying to break out of the system.

The first screen a student would see after the test was started was the option to select one of two sessions: Session 1 – The PISA Tests and Session 2 – The PISA Questionnaires. After selecting the appropriate session (which usually was done by the test administrator before the students arrived), the student was prompted for a login ID and password. The login ID was the 15 digit student ID assigned by KeyQuest as part of the sampling process. The password was also assigned by KeyQuest and was an 11 digit number. The first five digits comprised a checksum of the student ID, guarding against input errors. The next three digits encoded the test form which should be used for the student. The last three digits were a checksum of the three digit test form number.

After logging in, the student could optionally be shown a screen asking to select a language for this session. While the SDS was built with all the national languages available for a given country, it could be configured to support only one language. This was the recommended method of operation, where the test administrator chose the language configuration when starting the SDS, based on the school where the testing occurred. However, in some countries, it was necessary to allow the students to choose the language of assessment. The typical reason for allowing student choice for the language was for countries and schools with mixed language environments. In these cases, students decided at the time they started the survey session which language they wanted to use. The test administrator would guide students through the login and language selection process.

An important facet of the USB setup was protecting the test content on the USB drives. The PISA tests contain secure test materials, and people who obtain a USB drive should not have access to the test items except during the administration of the survey. To accomplish this, the files for rendering all test materials were stored in a MySQL database on each USB drive. The

files were stored in an encrypted format, and access to these was controlled via the web server. When a testing session was first started, the PISAMenu program would prompt for the password used to encrypt the files. Each country was assigned a unique password. This password was validated against known encrypted content in the database and then saved for the duration of the testing session. When a request was made to the web server for some part of the test content (e.g., one of the web pages or graphic images), the web server retrieved the content from the database and decrypted it on the fly.

One advantage of the SDS architecture used in 2018 was that it could be run without administrator rights to the local computer. This was a big improvement over past PISA cycles, reducing greatly the amount of technical support needed within the schools.

## DATA CAPTURE AND SCORING STUDENT RESPONSES

Student responses and other process data from the PISA tests and questionnaires were stored on the USB drives. Data was saved as the students answered each question, then exported at key intervals during the sessions. At the end of a session, the results from that session were exported in a single password protected ZIP file. For the PISA tests from Session 1 (the cognitive PISA domains, including the optional financial literacy domain), the ZIP files contained XML formatted data including logs of the students' actions going through the tests and files with the "variables" exported from the test. The following set of variables were exported for each item in the tests:

- Response: A string representing the raw student response.
- Scored Response: The code assigned to the response when the item was coded automatically.
- Number of Actions: The number of actions taken by the student during the course of interacting with the item. Actions counted were clicks, double-clicks, key presses and drag/drop events.
- Total Time: The total time spent on the item by the student.
- Time to First Action: The time between the first showing of the item and the first action recorded by the system for the item.

In addition to these five standard variables, some more complex items had custom variables that were of interest to the test development and psychometric teams. For instance, for the science simulations, the system exported counts of the number of experiments performed and the final set of results from each of these experiments.

An important task in PISA 2018 was coding of student responses. For computer delivered tests, many of the item responses could be coded automatically. In PISA 2018, this included multiple choice items, drag and drop items, numeric response items, and complex responses to science simulations.

For standard response modes, such as multiple choice or numeric entry, automated coding was done using a rule-based system. The correct answer (or partially correct answers in the case of partial credit items) were defined based on Boolean rules defined in a custom syntax. Simple conditionals were possible, e.g., to support different combinations of checkboxes in a multiple selection item where two out of three correct options should be selected. For numeric response items, the rules could check for string matches, which required an exact match against a known correct answer, or numeric matches, which used numeric equivalence to check an answer. For

numeric equivalence, for instance, 34.0 would match 34, but they would not match when using string matching.

A challenging part of evaluating numeric responses in an international context like PISA is how to parse the string of characters typed by the student and interpret it as a number. There are differences in decimal and thousands separators that must be taken into account, based on conventions used within countries and local usage. Use of these separators is not always consistent within a country, especially with increased migration and the pervasiveness of the Internet. For PISA 2018, the coding rules tried multiple interpretations of the student response to see if one of them could be coded as correct. The numbers were parsed in different ways, changing the decimal and thousands separators, testing each option to see if a correct response could be granted full or partial credit. Only if no alternate interpretation of the response resulted in a correct answer would the answer be coded as incorrect.

## OPEN ENDED CODING SYSTEM

While automatically coded items formed a significant portion of the units for PISA 2018, approximately 30% of the items required a response that needed to be coded by a human scorer or coder. On paper, this would be done directly on the test booklets. On the computer, a procedure was necessary to extract the responses provided by the students and present them to human coders. It was important to present these responses in a way that reflected the students' intent. This task is complicated by the fact that these responses could be more than just text. For example, for some items, a student would be required to select an option from a multiple choice part, then type in an explanation for why they chose that option. Additionally, in mathematics, students could use an equation editor to insert complex mathematics notation into their response.

For PISA 2018, the coding of these responses was done using the open-ended coding system (OECS). The OECS is a computer tool that was developed to support the coders in their work to code the responses according to the coding guides. All PISA 2018 open-ended responses collected with the computer-based platform were coded using the OECS.

The OECS works online so it required coders to have a reliable network connection. The OECS organizes responses according to the coding designs for each of the assessment domains. The system gives coders access to all the responses assigned to the coder. For each response, the coder will have access to part of the question for reference, the individual response to be coded, and the acceptable codes for each question. The coder selects the appropriate code and clicks on the "Record Code" button to save the selected code. It should be noted that this system was only used for response data from the computer-based assessment.

Also included on each page of the OECS were two checkboxes labelled "recoded" and "defer." The recoded box was used when the response had been recoded by another coder. The defer box was used when the coder was not sure what code to assign to the response. These deferred responses were typically reviewed and coded by the Lead Coder, or by the coder after consultation with the Lead Coder. When deferring a code for a response, coders were encouraged to enter comments into the box labelled "comment" to indicate the reason for deferring.

The OECS included the necessary features to support the monitoring of reliability. It organized all anchor, multiple and single coding of responses. According to a predetermined design, some

responses were single coded – coded by one person only – while others will be multiple coded – coded by more than one coder. Anchor responses (in English) were used to assess reliability across countries. Since the OECS gives coders only those responses that are assigned to them, coders do not know whether they are single or multiple coding. Once coding is complete for each item, the data can be integrated across coders and the OECS will generate reliability reports that includes multiple sections such as i) a summary, ii) item overview, iii) coders overview, iv) proportion agreement, v) coding category distributions, and vi) deferred and uncoded report.