



18

Computer-Based Assessments

Item rendering	390
Online item review	390
Translation	391
School computer requirements	391
Test delivery system	392
Data capture and scoring student responses	393
Online Coding System	393



PISA 2012 included three computer-based assessments (CBA):

- problem solving (PS);
- mathematics (CBAM); and
- digital reading (DRA).

Problem solving was undertaken by 44 countries. The computer-based mathematical literacy assessment and the digital reading literacy assessment were combined into a single international option and this option was undertaken by 32 countries (see Table 18.1).

The number of locales in Table 18.1 refers to the number of unique country by language combinations that were actually administered. A locale refers to a country by language version of the test. For example, Australian English, Canadian English, Canadian French and French French are four locales in two languages spread across three countries. The number of locales presented in this table is slightly different from the one presented in Chapter 13 that only refers to locales that participated in the coding reliability exercise.

Table 18.1 Number of countries and locales in PISA 2012 computer-based assessment

	Number of countries	Number of locales
Problem solving only	12	16
Problem solving, mathematics and digital reading	32	51

Chapter 2 dealt with the associated test development activities, test design and framework coverage of the computer-based assessment. This chapter focuses on the technicalities and functionality of the delivery system and the various supporting systems.

ITEM RENDERING

Item authoring was a collaboration between three professional groups: assessment item writers, graphic artists and JavaScript® programmers. The activities of the assessment item writers are described in Chapter 2. Graphic artists illustrated the ideas of the writers, using a variety of images and where free drawing was required, a consistent colour palette and illustrative style. To aid readability and focus the viewer, the overall design layout was kept clean using minimal extraneous elements, dark colours and gradients that distract from relevant content.

The problem solving and computer-based mathematics items were authored for PISA 2012 in JavaScript®. The DRA items were first used in PISA 2009, when they were Flash based. For PISA 2012 the DRA items were regenerated in JavaScript®.

JavaScript™ enables the programmer to vary the appearance and position of HTML elements that are displayed in a browser. Images, text and other HTML elements can be moved, changed in colour or size, hidden or revealed in response to user input such as mouse dragging, clicking or data entry. With its ability to manipulate HTML elements, JavaScript® effectively becomes a relatively strong two-dimensional animation tool.

Many of the mathematics and problem-solving items developed for PISA 2012 required some form of animation. This included actions such as dragging around the bars on a graph, highlighting a route on a map or changing the text and colours of a simple-state machine.

The use of JavaScript® in PISA 2012 avoided the problems experienced with the Flash-based solution in 2009: the input of right to left languages was not supported; the input of Cyrillic script was not supported; and the input of characters with diacritics was not fully supported.

ONLINE ITEM REVIEW

The item review activities described in Chapter 2 were conducted using a secure online review system developed by the international contractor. Each National Centre was provided with one primary account to securely view, rate and comment upon each item. Several secondary accounts (as many as requested) were provided so that national experts could securely view, rate and comment upon each item. The primary account contained a reporting facility that enabled the National Project Manager to view the responses from national experts and collate these into a single response per country through the primary account.



TRANSLATION

Only English source versions of the items were released for translation (or adaptation in the case of English testing locales). The workflows of the translation and verification processes were facilitated with an online Translation Management System (TMS) developed by the international contractor.

Translation of material was achieved by first creating XML (Extensible Markup Language) Localisation Interchange File Format¹ (XLIFF) compliant text files, and then translating or editing these files in an XLIFF compatible translation editor. The translation editor was a modified version of the open source Open Language Tool (OLT) used in PISA 2009. DIPF with the input of cApStAn modified the OLT primarily to deal better with right to left languages, customise it to the PISA workflow, and eliminate major known bugs.

The Translation Management System supported the translation and linguistic quality assurance workflow described in Chapter 5.

Once translation was completed, national language versions of the test delivery software were provided to National Centres via downloads from the international contractor's FTP sites in both Australia and Germany.

SCHOOL COMPUTER REQUIREMENTS

The basic hardware requirement for delivering the test was the availability of a suitable computer for each student. To be suitable, a computer needed to satisfy the following three criteria:

- run on Windows®, XP® Windows Vista or Windows 7® operating system;
- have a colour display with resolution of at least 1024 x 768 pixels; and
- have at least one accessible USB port (e.g. at the front of the machine).

More specific requirements, including processing power, system memory and data transfer rates, were verified at the time of system diagnosis (for specifics, see the section on system diagnostic below). The delivery application and results storage were on a single USB flash drive. The delivery application had to be opened with administrative privileges.

The computers had to be located so that the test could be supervised by a single Test Administrator, and in such a manner that students could not easily observe each other's screens.

System diagnostic

To determine the suitability of a computer for delivering the computer-based assessment, a system diagnostic tool was distributed by the international contractor. The PISA 2012 Hardware Diagnostic was designed in part to emulate the test delivery system, but it also provided feedback on the computer's memory, processing power and screen resolution. The minimum requirements are detailed in Figure 18.1.

■ Figure 18.1 ■

Minimum requirements, as checked by the PISA 2012 system diagnostic

System area	Requirement for CBA
CPU speed installed	1 500 Mhz
Operating system	Windows XP®, Windows Vista® or Windows 7®
The memory installed on the computer in MB	Windows XP® = 512 MB Windows Vista® = 1 024 MB Windows 7® = 1 024 MB
Available memory	Optimal requirements: 358 MB for Windows XP® 717 MB for Windows Vista® 717 MB for Windows 7® ----- Minimum requirements: 307 MB for Windows XP® 614 MB for Windows Vista® 614 MB for Windows 7®
Screen resolution	Minimal screen width = 1 024 px Minimal screen height = 768 px
Ensure Skype is not running on the computer	The CBA assessment cannot run with Skype™ running
The USB key data transfer rate	A minimum of 9 MB/s



As can be seen in Figure 18.1, the system diagnostic also checked that Skype™ was not running on the PC, as this interfered with the assessment display (Skype™ would recognise phone numbers in the assessment text and hyperlink them to Skype™ calling).

The system diagnostic also offered a virus check (of the USB, of the system memory, and of all available local and network drives). ClamWin™ Antivirus was used for this.

An optional advanced diagnostic check was also available. This check wrote a text file to the USB detailing as much relevant information as could be gathered about the system. The advanced check was useful where there was an issue preventing the optimal running of the assessment on a set of school computers but where all minimum requirements were met. The text file was sent to the international contractor for analysis to help determine the cause of any problem.

The system diagnostic interface source text could be localised – that is, replaced with text in the local language into which the assessment was translated.

TEST DELIVERY SYSTEM

The PISA 2012 computer-based assessments were delivered in schools via a set of software programs (described below) and national versions of the items, all bundled together onto a USB drive.

Generally, three variants of data collection were used by National Centres, sometimes in combination:

- the computers that existed in the sampled school were used to collect the data;
- laptops with preloaded software were carried into schools and used to collect the data; and
- students were transported to test centres.

An open-source computer-based assessment platform, TAO™ (*Testing Assisté par Ordinateur* or computer-based testing),² was used to sequence and store the items, store the results data, facilitate the student navigation, and provide all interface elements such as indicating progress through the test.

In PISA 2009 a Linux® OS based bootable system model was used to deliver a computer-based assessment. This model presented several problems, notably font recognition issues, drive recognition issues and the limitation of input methods to those recognised by Linux® (this was particularly frustrating for some Japanese and Chinese students). To overcome these problems in PISA 2012, the international contractor developed a Windows® OS based portable application.³ The major advantages of the Windows application came from its use of the native operating system: hence there were no font recognition or rendering problems, no hardware driver recognition issues, and students could use their native input methods.

There were some disadvantages to using a Windows® application. Firstly, it could only be used on computers with a Windows® operating system, thereby excluding Apple® and Linux® based operating systems which in some countries were popular. Secondly, being an external application, the Windows® User Management System, which is embedded into all Windows® operating systems, required that the application was run with administrative privileges. This “administrative rights” issue imposed by Windows® was a significant constraint in some schools and systems, as it was not possible to circumvent the Windows® User Management System. Where the school’s IT infrastructure is managed externally (e.g. by a contracted company or at the school district level), obtaining administrative permissions can be very difficult.

The Windows® application was basically a launcher for the test. In addition it was a launcher for the diagnostic tool that assessed the host computer for suitability for the test (see above section on system requirements).

Launching the test involved checking the computer for an available port, configuring a portable version of Firefox® to use that port, configuring the Apache™ web server and PHP (Hypertext Preprocessor) application server to run from the encrypted drive on the USB and then starting Apache™. Once Apache™ had started, the portable version of Firefox® was also started and opened to the entry page for the test. At this point the candidate was prompted to log in for the test to begin.

Access to the secure test was granted through a PHP login script. The student entered a 13-digit identifier (unique to each PISA student within a country). This identifier was then validated by entering a five-digit checksum. The checksum



was generated by applying a CRC (Cyclic Redundancy Check)¹⁶ security algorithm to the identifier. The identifier and checksum were communicated to the student via a form produced by the *KeyQuest* student sampling software. A second two-digit identifier (incorporating a checksum, also provided by *KeyQuest*) was entered by the student and this triggered automatic allocation of the appropriate pre-determined test form.

DATA CAPTURE AND SCORING STUDENT RESPONSES

Results data were written to the USB at regular intervals throughout the assessment. Data were transferred from National Centres to the international contractor via a secure FTP account.

In addition to results data that contributed directly towards students' cognitive scores, the system collected behavioural data such as time spent on task, sequence of pages visited, and use of stimulus elements such as drop down menus. In some cases, such behavioural data contributed to scoring.

All results data were stored in a single password protected ZIP file per student. Each ZIP file contained a set of XML files with each student's activities stored as a list of events. Of the 101 items across 3 domains (mathematics, problem solving and reading), 83 items were required to be automatically scored; each of these items was associated with a specific scoring algorithm. Event data were extracted from the ZIP files and processed with the following data being retained for further analysis: student details, item details, raw data log, response contents, score, time spent on each item. Auto-scored items include multiple choice, complex multiple choice, numeric input, dropdown select and many of the interactive tasks. The remaining open constructed-response items were imported into the Online Coding System, developed by the international contractor, to be coded by experts trained within each National Centre.

ONLINE CODING SYSTEM

The user interface for the Online Coding System was localisable – the language elements could be translated into any language variant used in the computer-based assessment. The National Project Manager was able to create user accounts with different roles:

- a coding supervisor responsible for organising and managing the online coding operation;
- leading coders, who played a central role in monitoring the quality of coding, as well as coding responses themselves; and
- coders.

The quality of the coding was monitored in several ways. For the more complex Constructed-Response items, multiple coding was applied to a subset of responses. Each multiply-coded response was coded by a minimum of four coders with each coder being allocated 100 responses for multiple coding when the assessment language was a major verified language, and 50 responses when the assessment language was a minor verified language. The remaining responses for these complex Constructed-Response items were assigned randomly to the coding group and coded once.

All responses to simple Constructed-Response items were coded once.

The coding assignment was handled by the online coding software, however the supervisor had the ability to determine a minimum proportion of responses to be allocated to each coder. This ensured that slow or absent coders still coded a reasonable number of responses, and that one or two fast coders did not code all the responses. Spreading the coding load among at least four coders reduced the risk of bias being introduced by an individual's harshness, leniency or misunderstanding of the coding guide. The method of reliability analysis and the results are described in detail in Chapter 13.

Another quality assurance measure was the introduction of control scripts for some items. These control scripts were translations of standard responses, to which the code to be applied was known to the international contractor. Control scripts were inserted into the coding workflow, blind to the coder, and analysed by the international contractor to determine whether codes had been applied correctly and consistently across countries. The method and results of this analysis are described in Chapter 13.

During the coding of the item, the Leading Coder spot-checked the work of coders each day. Spot-checking involved a review of codes assigned to responses. A general guide was that at least 10 responses per item were spot-checked.

If a coder was uncertain about the code to assign to a particular response, the response could be marked for review and it would be sent automatically to a leading coder or supervisor for advice.



The Online Coding System provided several reports to help the coding supervisor manage the quality and workflow of the coding process, including:

- a short summary of allocated items listing the coders assigned to that item, the number of responses remaining for each coder, the number of single-coded responses that have yet to be assigned and the total number of responses coded by each coder;
- a complete summary of allocated items which, in addition to the information in the short summary, contained data for each item on the minimum and maximum number of single-coded responses assigned to each coder, the total number of responses that had been multiple-marked and single-marked, and, when coding was complete for each item, item reliability statistics were made available; and
- a report listing any instances in which the missing code had been applied inconsistently to the same response.

Notes

1. For a description, see <http://en.wikipedia.org/wiki/XLIFF>. Version 1.1 was current and specifications for this version can be found at <http://www.oasis-open.org/committees/xliff/documents/cs-xliff-core-1.1-20031031.htm>.

2. Developed by the Centre de Recherche Public (CRP): Henri Tudor and the Université du Luxembourg. See <https://www.tao.lu/>.

3. Developed by the Deutsche Institut für Internationale Pädagogische Forschung (DIPF) in co-operation with the Centre de Recherche Public (CRP).