# 1. Introduction

## 1.1 Context

There is a desire by UNAIDS to encourage countries that have implemented DHIS2 to generate indicators that use Spectrum[[1]](#footnote-1) estimates.

It has been successfully demonstrated that the extracts generated from Spectrum can be imported into DHIS2. Furthermore, an analysis of the desired indicators revealed that certain components would be common across countries.

This document describes the requirements for implementing a new ‘UNAIDS bootstrapping app’ in DHIS2 that will allow countries to initialise a standardised set of data elements, indicators and dashboard items in their DHIS2 platform.

## 1.2 Simplifying the UNAIDS DHIS2 setup process

Since the import of Spectrum derived data is a ‘new’ feature that UNAIDS wants to encourage countries to adopt, there is an opportunity to specify a pre-defined set of common DHIS2 objects that can be applied to any instance of DHIS2.

The DHIS2 platform provides:

* The ability to export and import DHIS2 object metadata definitions; and
* The ability to create custom modules though the use of an App Marketplace

Given these features, it is possible to ‘bootstrap’ an instance of DHIS2 with pre-defined ‘place-holder’ objects and thus reduce the barrier for countries that want to leverage Spectrum derived estimates.

The following identified areas lend themselves to pre-defined initialization:

* The Spectrum data elements used for direct import of extracts
* The indicators that will use the Spectrum data elements (as denominators)
* Certain dashboard items that will leverage the Spectrum indicators

It should be noted that these placeholder objects will need further customisation once initialized due to the localized nature of a DHIS2 implementation.

## 1.3 What this document contains

This document provides the following:

* Narrative explanation of the application to be created
* A high-level list of requirements
* Appendices providing relevant samples of example DHIS2 metadata definition files

Developers can rapidly prototype the application, using the attached definition files. Final versions of the files will be provided as the project progresses.

# 2. The Narrative View

## 2.1 DHIS2 Background

The DHIS2 platform starts as a blank slate, where all components of a health information system need to be defined (e.g. facilities, data elements, indicators, dashboards, users, disaggregation metadata etc.).

DHIS2 has three core concepts around which it organizes its aggregate data collection: data elements, indicators and organizational hierarchies. The organizational hierarchy is typically used to breakdown a country into administrative areas, sub-areas and facilities, matched to the needs of the implementer (e.g. a Ministry of Health). Data elements are the atomic level items for holding data, while indicators are calculated items derived from data elements (e.g. prevalence for a certain disease).

Data elements and indicators are linked to organizational unit(s) and data is typically collected at the lowest level possible, and aggregated by the system to generate higher organizational level values for data elements and indicators. A data element may also be defined in such a way that it can be disaggregated into relevant sub-component parts (e.g. by gender and age-brackets).

Each object in DHIS2 (e.g. data element, user, indicator, disaggregation category, organization unit etc.) is allocated a GUID[[2]](#footnote-2) that is unique across all instances of DHIS2. When importing data into DHIS2, GUIDs can be used to precisely identify which objects a specific data item will be linked to.

## 2.2 Mixed Universal and Local Components

The overall requirement, which this bootstrapping application is attempting to simplify, is to allow a country to use a pre-defined set of indicators and dashboard elements (visualisations) that leverage Spectrum estimates. However, since these indicators will use both Spectrum estimates (a universally defined component) and data already collected in DHIS2 (the locally defined component) to derive a value, they can only be partially specified (further manual customisation will be necessary).

For example, one such indicator, the *‘Estimated Percentage of all people living with HIV who are receiving ART’*, will be made up of the numerator *‘Number of people currently on ART’* (an existing locally defined component) and the denominator *‘Estimated number of people living with HIV’* (a new universally defined component).

It should be self-evident therefore, that the initialisation of these common elements via a bootstrapping application will provide a ‘skeleton framework’ only (placeholders), which the implementing country will further customise using local components (and a set of guidelines).

## 2.3 The UNAIDS DHIS2 Bootstrapping Application

The following narrative has been split between ‘should haves’ and the ‘could haves’. The ‘should haves’ define the minimum requirement, while, depending on time, the ‘could haves’ define less important but nice-to-have features.

### 2.3.1 The Should Haves

The bootstrapping application is designed to selectively define a set of common DHIS2 placeholder objects that need to be further customised with the assistance of a set of accompanying guidelines. Customisation will always be necessary due to the localised nature of DHIS2 implementations.

The bootstrapping application is relatively simple, as it only needs to provide the ability to load a set of pre-defined metadata definitions into DHIS2 using (preferably) the API.[[3]](#footnote-3) It is preferable (though not necessarily a ‘must’) that the metadata definitions be represented as files to allow new definitions to be added, or changes made to the initialization definitions with relative ease.

The bootstrapping application should selectively allow the user to specify 1, 2 or all 3 of the following initialisations (recognising that each is progressively dependent on the previous initialisation):

* The Spectrum data elements used for direct import of extracts (must exist for the following two initializations)
* The indicators that will use the Spectrum data elements (dependent on the previous initialization and must exist for the following initialization)
* Certain dashboard items that will leverage the Spectrum indicators (is dependent on the previous two initializations).

The ability to ‘progressively’ select 1, 2 or all 3 of the initialisations allows a country to determine the relative level of ‘common’ initialisation they want to adopt (the minimum being the common Spectrum data elements).

It is likely that most countries that initialise their DHIS2 instance using this bootstrapping application will select at a minimum the ‘data elements’ and ‘indicator’ initialisation options.

### 2.3.2 The Could Haves

The above level of functionality is relatively simple and can be rapidly prototyped. A further level of functionality would be the provision of access to the customisation guidance necessary to utilise the initialisations. These will consist of a set of written guidelines and videos that could either be provisioned on the local DHIS2 instance or alternatively refer to a global location. However, since international connectivity is not always consistent, locally availability of guidance materials is probably preferential.

Since the materials are still being produced, if this functionality is prototyped, it would simply provide access to a dashboard of relevant guidance materials (that could also be defined via a DHIS2 dashboard metadata file).

# 3. High-level Requirements

The following provides a more formal representation of the requirements of the UNAIDS DHIS2 Spectrum Bootstrapping Application.

### REQ-1: The application must be able to initialize a set of DHIS2 objects based on a definition specification

This is THE key requirement.

|  |  |
| --- | --- |
| **Details** |  |
| **Parent**: DHIS2 | |
| **Type:** Functional | **Status:** |
| **Assigned To:** App Developers | **Release:** |
| **Notes** | **Date Added** |
| Since this is a standard capability of DHIS2, the App must be able to leverage the API to initiate the creation of a pre-defined set of DHIS2 objects. | 01/06/2015 |
| **Use cases that reference this requirement** | |
|  |  |

#### REQ-1.1: The App should be able to use files to represent the DHIS2 metadata object definitions

Metadata definition files can be exported or imported to and from DHIS2 for any object or set of objects. The preferred file format is XML, and since a ‘representative’ DHIS2 instance has been used to model the various object placeholders for UNAIDS Spectrum derived indicators, these form the basis for the definitions to be initialized by the bootstrapping app.

|  |  |
| --- | --- |
| **Details** |  |
| **Parent**: REQ-1: The application must be able to initialize a set of DHIS2 objects based on a definition specification | |
| **Type:** Functional | **Status:** |
| **Assigned To:** App Developers | **Release:** |
| **Notes** | **Date Added** |
| While the developers may decide to implement this functionality differently, the use of files (one for each area of definition) to hold the definitions will allow a more maintainable application. Definition files can also be easily created by DHIS2 and therefore the scope of development required significantly reduced if these files are adopted as the definition container.  XML extract examples have been provided in the appendices. | 01/06/2015 |
| **Use cases that reference this requirement** | |
|  |  |

#### REQ-1.1.1: The App should be able to specify (and retain) the location of the definition files via a ‘preferences’ setting interface

While not normally changed, the default locations and names of the definition files should be able to be specified and retained by the bootstrapping app. A minimum of 4 options should be defined:

* Location and filename of the Data Elements definition file(s)
* Location and filename of the Indicators definition file(s)
* Location and filename of the Dashboard definition files(s)
* Location and filename of the Guidance Dashboard definition file(s)

|  |  |
| --- | --- |
| **Details** |  |
| **Parent**: REQ-1.1: The App should be able to use files to represent the DHIS2 metadata object definitions | |
| **Type:** Functional | **Status:** |
| **Assigned To:** App Developers | **Release:** |
| **Notes** | **Date Added** |
| While the developers may decide to implement this functionality in a number of ways, a common approach with most applications is to use a ‘preferences’ or ‘options’ admin panel that allows the user to change the defaults for these.  Warning messages should be used to highlight that changes to the default location/names of the files may result in a failure of the intended initialisation functionality. | 01/06/2015 |
| **Use cases that reference this requirement** | |
|  |  |

### REQ-2: The application should be able to allow user selection of the level of initialisation to implement

There are to be three general areas of initialization. For each area, a separate metadata definition will exist (probably represented as XML files as per REQ 1.1). The user should be able to select 1, 2 or all 3 areas for initialization. The areas include:

* Spectrum Data Elements
* Spectrum Derived Indicators
* Spectrum Derived Dashboards

The three areas are progressive, with each level dependent on the previous. Therefore, the application should include logic to restrain initialisation of latter options if previous options are not selected.

|  |  |
| --- | --- |
| **Details** |  |
| **Parent**: DHIS2 | |
| **Type:** Functional | **Status:** |
| **Assigned To:** App Developers | **Release:** |
| **Notes** | **Date Added** |
| The user interface could be implemented via ‘option boxes’ with dependent options ‘greyed out’ until prior options are selected. Selection of the given option(s) will queue the related definition file for initialisation.  An ‘initialise’ button should be greyed out until one or more options are selected. Once ‘clicked’ the application will load the appropriate definition file, provide a ‘progress’ indicator, and a ‘completed’ status when the initialisation is complete. | 01/06/2015 |
| **Use cases that reference this requirement** | |
|  |  |

#### REQ-2.1: The App should allow the user to select a sub-option under the main option Spectrum Data Elements

Since country customization of a given DHIS2 instance usually involves the generation of local ‘age-bracket’ and ‘gender’ definitions (and related object UIDs), it is not possible to universally apply these to generated data elements via this application.

The spectrum extracts will – for some indicators – provide disaggregation for <15 / 15+ age brackets (by gender). In some instances, countries will already have these age brackets, and in some cases they will not. For those countries that do not have these age-bracket already defined, they have the option of initializing these age-brackets via this App (they will be needed to complete the customization process of the data elements as indicated in the accompanying Guidance documents).

The requirement therefore is to allow the addition of this age-bracket setup as part of the Spectrum Data Elements initialization option. It is simply a small fragment of XML that needs to be applied in addition to the main Data Elements definition file. It could be implemented as two separate versions of the Data Elements definition file, or as a single Data Elements definition file plus XML fragment (the former approach being more maintainable).

|  |  |
| --- | --- |
| **Details** |  |
| **Parent**: REQ-2: The application should be able to allow user selection of the level of initialisation to implement | |
| **Type:** Functional | **Status:** |
| **Assigned To:** App Developers | **Release:** |
| **Notes** | **Date Added** |
| The user interface could simply be added as part of the same options panel as outlined for REQ 2. When the Spectrum Data Elements option is ‘ticked’, a ‘greyed-out’ sub-option for ‘Apply <15/15+ age brackets’ could become active for selection. | 01/06/2015 |
| **Use cases that reference this requirement** | |
|  |  |

### REQ-3: Expose a Guidance Dashboard (Could Have)

Guidance documents and other artifacts can be loaded into a DHIS2 instance, and also made visible via a ‘dashboard’. This is the preferred method for implementing ‘guidance’ for customizing the initialized components with localized information. The initialization of a Dashboard using the DHIS2 XML definition capability allows the App to be ‘lighter’ and much more maintainable (changes to the dashboard, definitions and guidance specifics can be done without having to modify the App).

The first time the App is used to initialize any of the Spectrum components, the user should be prompted with a ‘preferred language’ option for the guidance documents. It should be noted that the App would need to ‘load’ an appropriate set of guidance files into the DHIS2 instance (preferably from a global location to ensure the latest version of guidance is available).

|  |  |
| --- | --- |
| **Details** |  |
| **Parent**: DHIS2 | |
| **Type:** Functional | **Status:** |
| **Assigned To:** App Developers | **Release:** |
| **Note** | **Date Added** |
| The Guidance Dashboard will be initialized in a manner similar to the other definition components, but will be triggered automatically if the user selects 1 or more of the initialization options.  The Guidance Dashboard will only be visible to s specific user setup for this purpose (defined in the Guidance Dashboard definition file).  A message should be displayed therefore as part of initialization with the user ID/Password. Alternatively, the Guidance Dashboard could appear to form part of the App itself (e.g. as a separate TAB within the App interface). | 01/06/2015 |
| **Use cases that reference this requirement** | |
|  |  |

### REQ-4: Status, Audit and Synchronization Functions (Could Have)

The application should keep an audit trail of how many times it has been run, which initializations have already been run, and the status of any non-volatile items that need to be stored (such as preferences for locations of files, file names etc.).

A global preference option could be to turn on ‘synchronization’. In such a case, a central globally maintained repository of Spectrum and Guidance definition files can be referenced and asynchronously (i.e. when connectivity permits) checked for updates (e.g. new data elements, indicators or dashboard items). The local versions of these files would be updated as necessary as a background process.

|  |  |
| --- | --- |
| **Details** |  |
| **Parent**: DHIS2 | |
| **Type:** Functional | **Status:** |
| **Assigned To:** App Developers | **Release:** |
| **Note** | **Date Added** |
| While this is a could-have, the introduction of this functionality (provided time permits) would allow the App to be universally maintainable. | 01/06/2015 |
| **Use cases that reference this requirement** | |
|  |  |

# Appendix 1 – DHIS2 Data Elements XML definition sample

# Appendix 2 – DHIS2 Indicator XML definition sample

# Appendix 3 – DHIS2 Dashboard XML definition sample

1. Spectrum – the software package used to model HIV/AIDS estimates [↑](#footnote-ref-1)
2. A universally unique ID known as a UID in DHIS2 [↑](#footnote-ref-2)
3. Application Programming Interface [↑](#footnote-ref-3)