

# User Manual: piql EARK eHealth SIP Creator

## Table of Contents

EARK eHealth SIP Creator .....	2
1.0 Introduction .....	2
2.0 Descriptive Metadata .....	3
2.0 Extracting Patient records from EMR Systems .....	4
2.1 Assumptions and Limitations .....	4
2.2 EMR Export Structure .....	4
3.0 Transformation Scripts and Automation .....	6
4.0 Using the SIP Creator .....	7

# EARK eHealth SIP Creator

Intended audience: Archivist, Hospital Administrator, Hospital Database Administrator

## 1.0 Introduction

The Piql EARK SIP Creator is a version of the Windows desktop 'piqlIngest' tool that is designed to create conformant EARK eHealth1 SIPs from exports of Electronic Medical Record (EMR) systems which are structured in a prescribed manner. The tool can also automate bespoke scripts that transform non-conformant exports into conformant exports by mapping and renaming directory structures and by mapping metadata files.

Patient data as submitted by hospitals or healthcare providers is likely to be periodically extracted from source systems and sent in batches to the central health archive. The eHealth1 specification allows for the inclusion of multiple patients per package and so these batches can be transmitted in a single submission. The number of patient medical records contained within each archival package (AIP) is a matter for local implementation at the Central Archive. The eHealth1 SIP Creator is delivered as a self-contained zipped directory structure which contains executable code, all dependencies, reference metadata schemas and sample data.

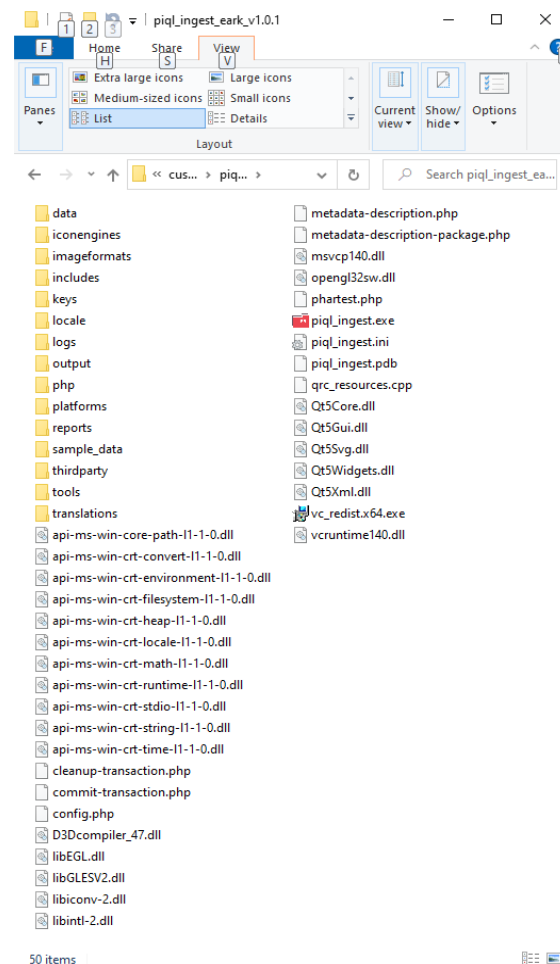


Figure 1 : eHealth1 SIP Creator Package Contents

## 2.0 Descriptive Metadata

Controlled vocabularies and coding provide a standardised way for unambiguous recording of health data. Most EMR and all EHR systems will hold coded data concerning patient records or journals that can be extracted as metadata for the Patient Medical Record and will use international standards such as ICD or SNOMED. Data can be recorded in a number of standardised formats (such as HL7 FHIR) or to a local format which is specified by the health archive and referenced within a Submission Agreement.

The eHealth1 CITS Specification requires that Patient Personal descriptive metadata **MUST** be included in the information package at the root level and that Patient Clinical information **MAY** be included within each Patient Medical Record. The FHIR Patient and condition resources are recommended but not mandated and the SIP Creator has been configured to use these schemas as standard. At a minimum the Patient Personal descriptive metadata should be a simple patient manifest with patient names and references to unique identifiers (IDs).

eHealth1 SIPs should include all necessary xml schemas and the SIP Creator has been profiled to include schemas for: METS, METS extensions, xlink, xml and the fhir base, patient and condition schemas as shown in figure 2.

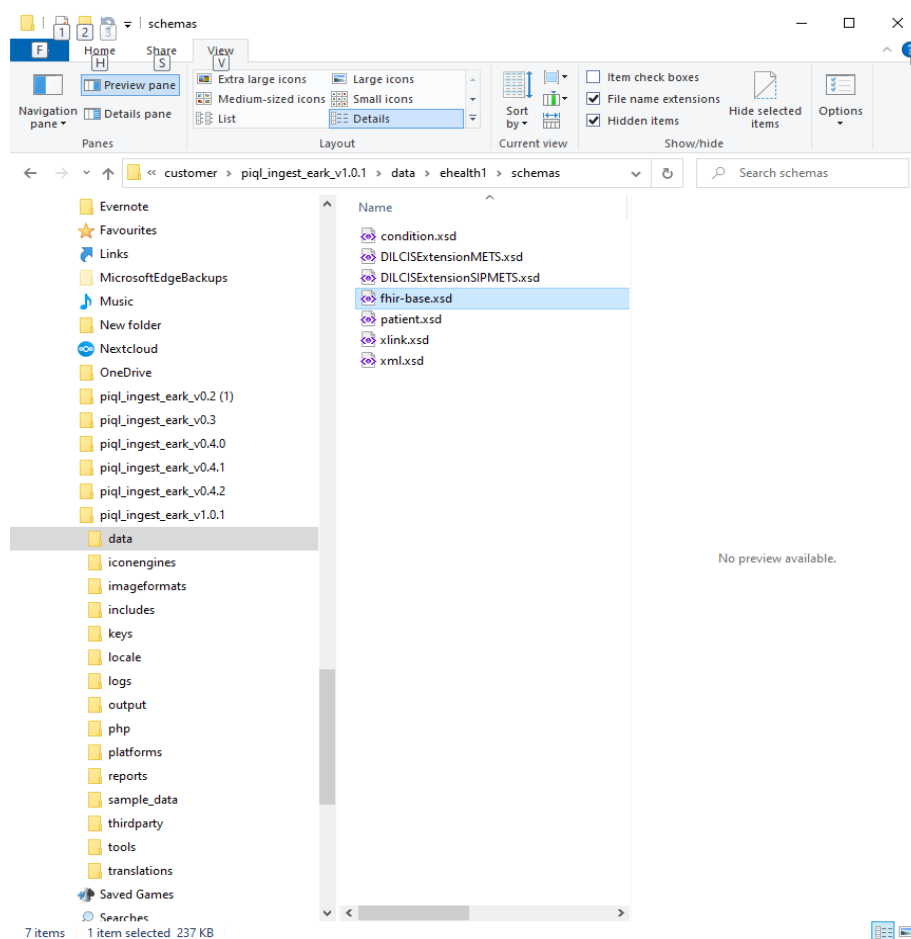


Figure 2 : Metadata Schemas

The use of fhir resources is not mandatory and these schemas can be exchanged for local standards. Schemas can also be referenced externally to the package (see 4.0 Using the SIP Creator).

## 2.0 Extracting Patient records from EMR Systems

### 2.1 Assumptions and Limitations

It is assumed that with the skills of a local DBA, basic scripting and database queries that that organisations will be able to:

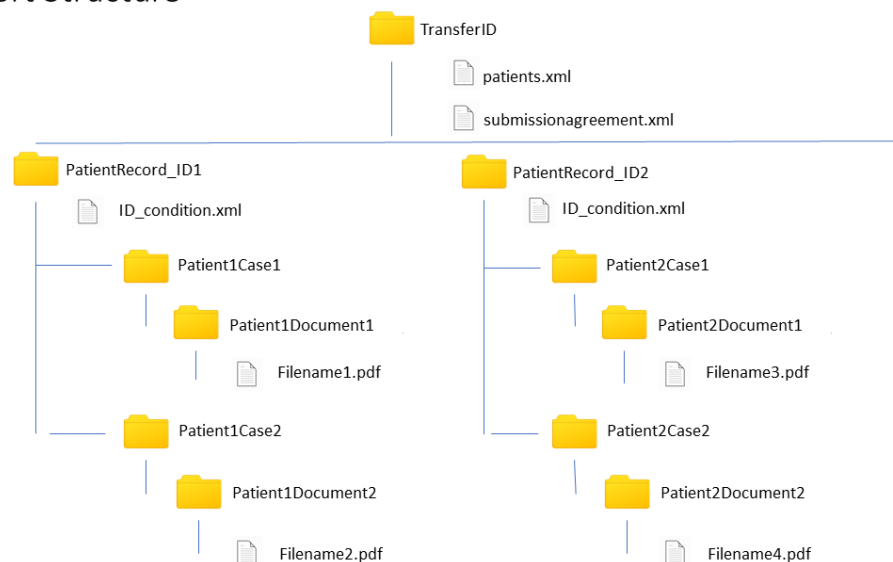
- Extract patient administrative metadata together with unique patient IDs and save to an xml file which will comply to either: the fhir-patient resource, or a local xml schema. This can be for multiple patients for each submission.
- Extract patient medical record information (folders, documents, files) as separate patient records, each organised according to one of the case structures as described in the eHealth1 specification section 6.3 (Example 1 - Entire Patient Medical Record as one file, Example 2 – Patient Medical Record as set of thematic files and Example 3 – Patient Medical Record as a set of Documents per Case).
- Extract minimum viable patient case clinical data that conforms to a recognised vocabulary (such as ICD), is linked to patient cases in one of the above structures and is saved to xml files that comply with either: fhir-condition resource, any other fhir clinical resource, or to a local xml schema. There should be references for each record to the relevant patient cases.

### 2.2 EMR Export Structure

It is assumed that with the skills of a local DBA, basic scripting and database queries that organisations will be able to extract data from source EMR systems as follows:

- Patient administrative metadata with unique patient IDs saved to a master xml metadata file at the root of a package folder structure. The schema for the xml master file should be specified, for example the fhir patient resource.
- Patient medical record and patient case information (documents, files) in one of the case structures as described in the eHealth1 specification section 6.3 (Example 1 - Entire Patient Medical Record as one file, Example 2 – Patient Medical Record as set of thematic files and Example 3 – Patient Medical Record as a set of Documents per Case. This patient and case structure must be represented as a physical folder structure containing files and contain a reference to the patient unique identifier (ID). Non-physical case structures can be mapped by scripts that can be automated within the SIP Creator.
- Minimum viable patient case clinical data that conforms to a recognised vocabulary (such as ICD), linked to patient cases in one of the above structures in an xml file with references to the patient ID and case IDs. Case IDs should be included in the case folder names.

## eHealth1 SIP Creator EMR Export Structure



## 2.3 Package Folder and File Naming

Naming conventions are required for input data (the EMR system extraction) as follows:

Item	Name	Location
Submission package name	Name of root folder (should contain a unique package identifier)	
Submission Agreement	submissionagreement.pdf	Root folder
Patient Personal Data	patients.xml	Root Folder
Patient Clinical Data	Prefixed by the Patient ID contained in the patients xml file and the patient record folder name and suffixed by 'condition' separated by underscore, e.g. ID_condition.xml	Root of each patient record folder
Patient Records	Organised in folders by Patient and prefixed by 'Patientrecord' and suffixed by the patient unique ID separated by underscore, e.g. patientrecord_ID	In Data Folder
Data Files	EMR system generated filenames	In case/sub-case/document folder structure in patient record folder
Case/sub-case folder structure	EMR system generated folder names which should be unique within each record and may contain a reference to the level (e.g. case, subcase, document). References to cases used in the clinical metadata should be included in the folder name.	In patient record e.g. patient1case1

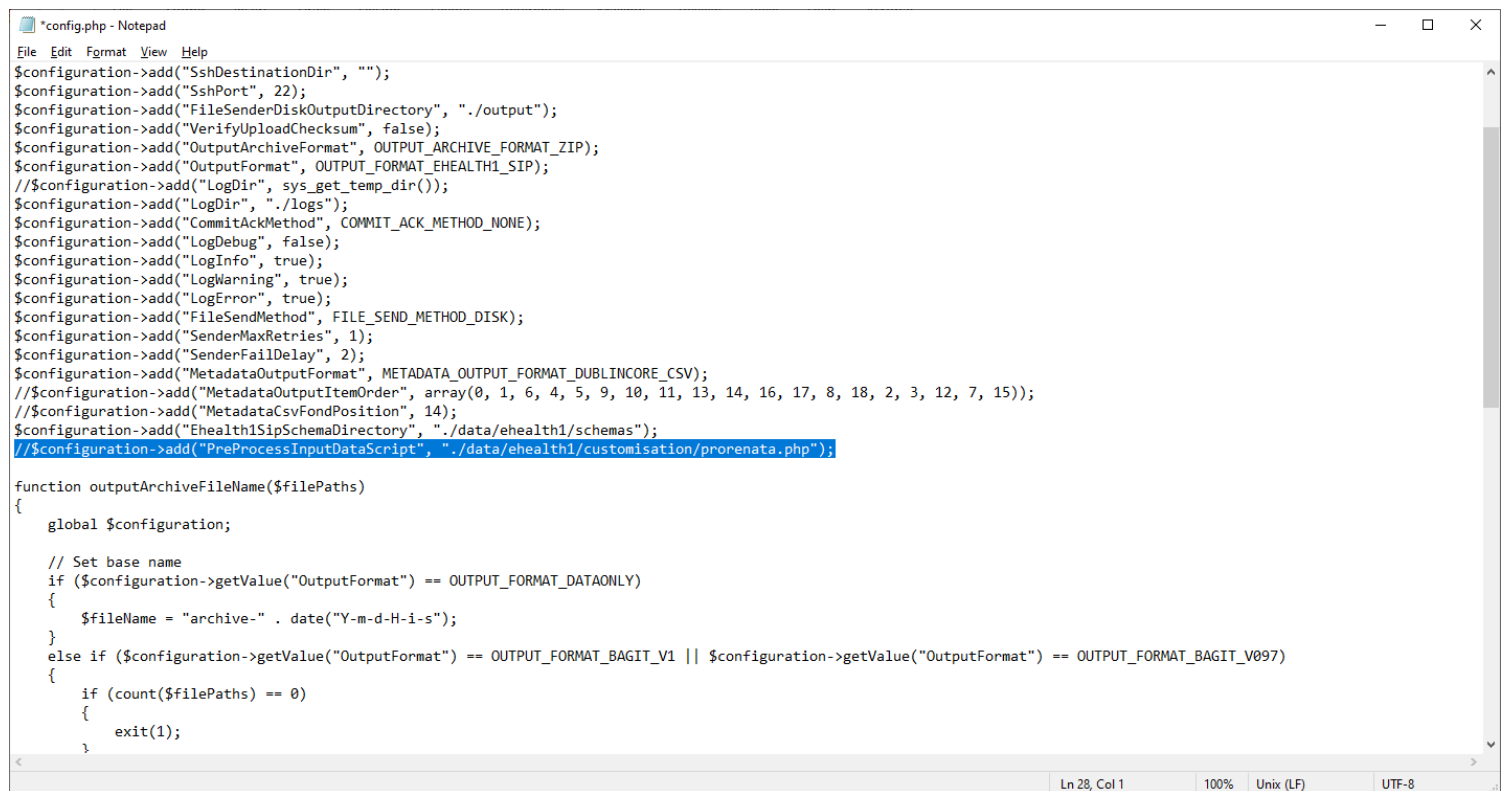
An example of a conformant EMR export, called eHealth1\_conformant\_transfer\_0, is included in the SIP Creator package in the /sample data folder. This uses the fhir patient and condition resources.

### 3.0 Transformation Scripts and Automation

The SIP Creator can be configured to map non-conformant EMR exports by means of scripts. These scripts can be automated within the SIP Creator workflow. The SIP Creator package contains an example of a non conformant export named eHealth\_nonconformant\_transfer\_0 in the sample data folder. An example script that transforms the eHealth1\_nonconformant\_transfer export into a conformant one is also included in the package in the folder /data/ehealth1/customisation/.

The use of a customisation script can be turned on within the config.php file within the root of the SIP Creator package. Figure 4 shows the config.php without customisation and figure 5 with customisation, where it can be noted that a transformation script has been enabled by removing the ‘//’ comment tags on line:

```
// $configuration->add("PreProcessInputDataScript","./data/ehealth1/customisation/prorenata.php");
```



```

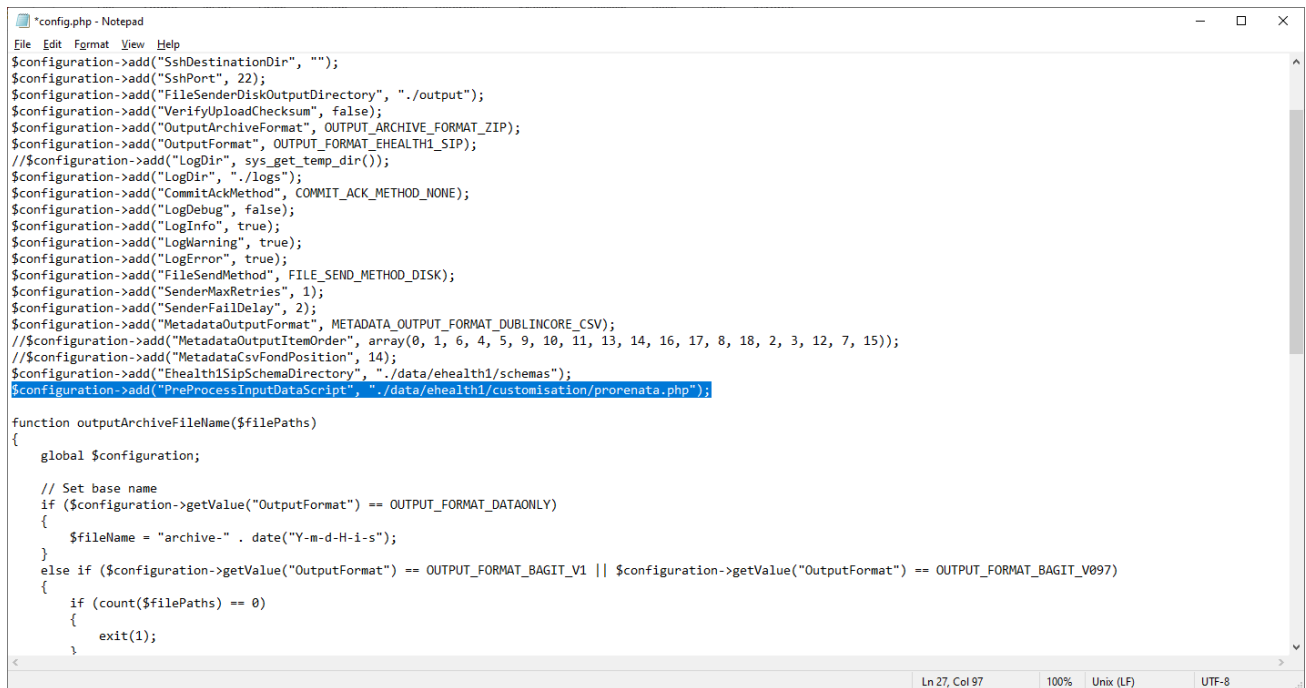
"config.php - Notepad
File Edit Format View Help
$configuration->add("SshDestinationDir", "");
$configuration->add("SshPort", 22);
$configuration->add("FileSenderDiskOutputDirectory", "./output");
$configuration->add("VerifyUploadChecksum", false);
$configuration->add("OutputArchiveFormat", OUTPUT_ARCHIVE_FORMAT_ZIP);
$configuration->add("OutputFormat", OUTPUT_FORMAT_EHEALTH1_SIP);
// $configuration->add("LogDir", sys_get_temp_dir());
$configuration->add("LogDir", "./logs");
$configuration->add("CommitAckMethod", COMMIT_ACK_METHOD_NONE);
$configuration->add("LogDebug", false);
$configuration->add("LogInfo", true);
$configuration->add("LogWarning", true);
$configuration->add("LogError", true);
$configuration->add("FileSendMethod", FILE_SEND_METHOD_DISK);
$configuration->add("SenderMaxRetries", 1);
$configuration->add("SenderFailDelay", 2);
$configuration->add("MetadataOutputFormat", METADATA_OUTPUT_FORMAT_DUBLINCORE_CSV);
// $configuration->add("MetadataOutputItemOrder", array(0, 1, 6, 4, 5, 9, 10, 11, 13, 14, 16, 17, 8, 18, 2, 3, 12, 7, 15));
// $configuration->add("MetadataCsvFondPosition", 14);
$configuration->add("Ehealth1SipSchemaDirectory", "./data/ehealth1/schemas");
// $configuration->add("PreProcessInputDataScript", "./data/ehealth1/customisation/prorenata.php");

function outputArchiveFileName($filePaths)
{
    global $configuration;

    // Set base name
    if ($configuration->getValue("OutputFormat") == OUTPUT_FORMAT_DATAONLY)
    {
        $fileName = "archive-" . date("Y-m-d-H-i-s");
    }
    else if ($configuration->getValue("OutputFormat") == OUTPUT_FORMAT_BAGIT_V1 || $configuration->getValue("OutputFormat") == OUTPUT_FORMAT_BAGIT_V097)
    {
        if (count($filePaths) == 0)
        {
            exit(1);
        }
    }
}
Ln 28, Col 1    100%    Unix (LF)    UTF-8

```

Figure 4 : config.php with customisation script disabled



```

*config.php - Notepad
File Edit Format View Help
$configuration->add("SshDestinationDir", "");
$configuration->add("SshPort", 22);
$configuration->add("FileSenderDiskOutputDirectory", "./output");
$configuration->add("VerifyUploadChecksum", false);
$configuration->add("OutputArchiveFormat", OUTPUT_ARCHIVE_FORMAT_ZIP);
$configuration->add("OutputFormat", OUTPUT_FORMAT_EHEALTH1_SIP);
// $configuration->add("LogDir", sys_get_temp_dir());
$configuration->add("LogDir", "./logs");
$configuration->add("CommitAckMethod", COMMIT_ACK_METHOD_NONE);
$configuration->add("LogDebug", false);
$configuration->add("LogInfo", true);
$configuration->add("LogWarning", true);
$configuration->add("LogError", true);
$configuration->add("FileSendMethod", FILE_SEND_METHOD_DISK);
$configuration->add("SenderMaxRetries", 1);
$configuration->add("SenderFailDelay", 2);
$configuration->add("MetadataOutputFormat", METADATA_OUTPUT_FORMAT_DUBLINCORE_CSV);
// $configuration->add("MetadataOutputItemOrder", array(0, 1, 6, 4, 5, 9, 10, 11, 13, 14, 16, 17, 8, 18, 2, 3, 12, 7, 15));
// $configuration->add("MetadataCsvFondPosition", 14);
$configuration->add("Ehealth1SipSchemaDirectory", "./data/ehealth1/schemas");
$configuration->add("PreProcessInputDataScript", "./data/ehealth1/customisation/prorenata.php");

function outputArchiveFileName($filePaths)
{
    global $configuration;

    // Set base name
    if ($configuration->getValue("OutputFormat") == OUTPUT_FORMAT_DATAONLY)
    {
        $fileName = "archive-" . date("Y-m-d-H-i-s");
    }
    else if ($configuration->getValue("OutputFormat") == OUTPUT_FORMAT_BAGIT_V1 || $configuration->getValue("OutputFormat") == OUTPUT_FORMAT_BAGIT_V097)
    {
        if (count($filePaths) == 0)
        {
            exit(1);
        }
    }
}

```

Figure 5 : config.php with customisation script enabled

## 4.0 Using the SIP Creator

The SIP Creator is launched by double clicking the executable file 'piql\_ingest.exe' at the root of the package (see Figure 1). The tool can be run with the package copied to a local disk drive or it can be run from portable media like a USB stick. The package should not be altered.

The tool will launch the window shown in Figure 6.

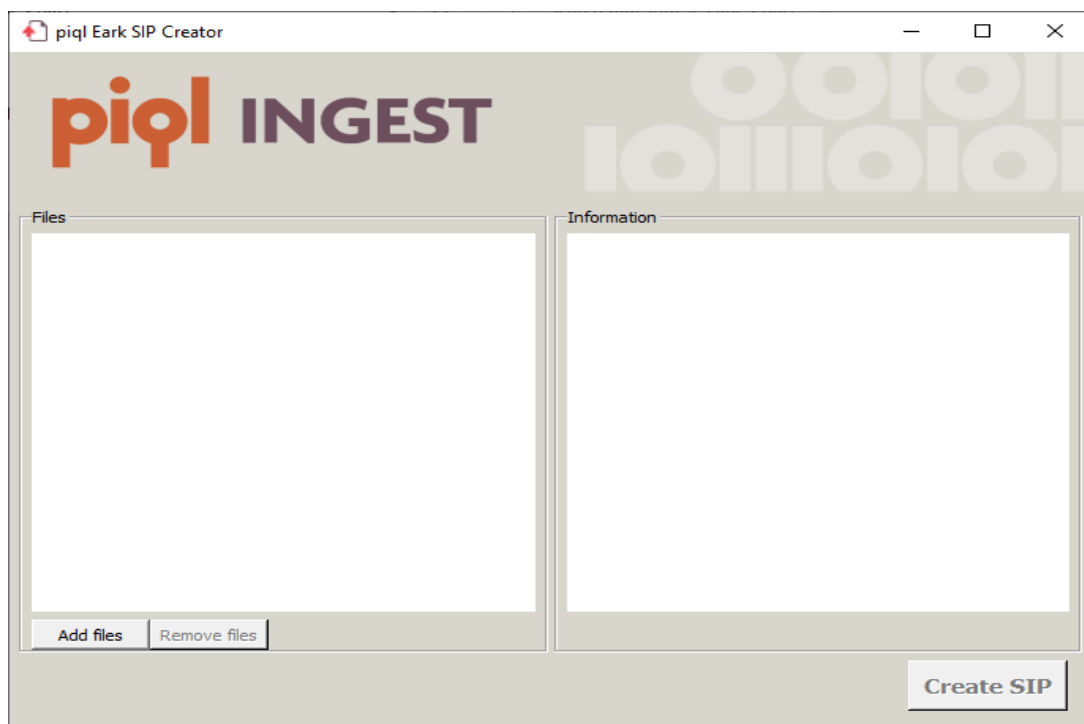


Figure 6 : SIP Creator Window

Submission packages are ingested into the tool by dragging and dropping from the top-level folder of the submission. See Figure 7 which is using the conformant test submission from the /sample data folder.

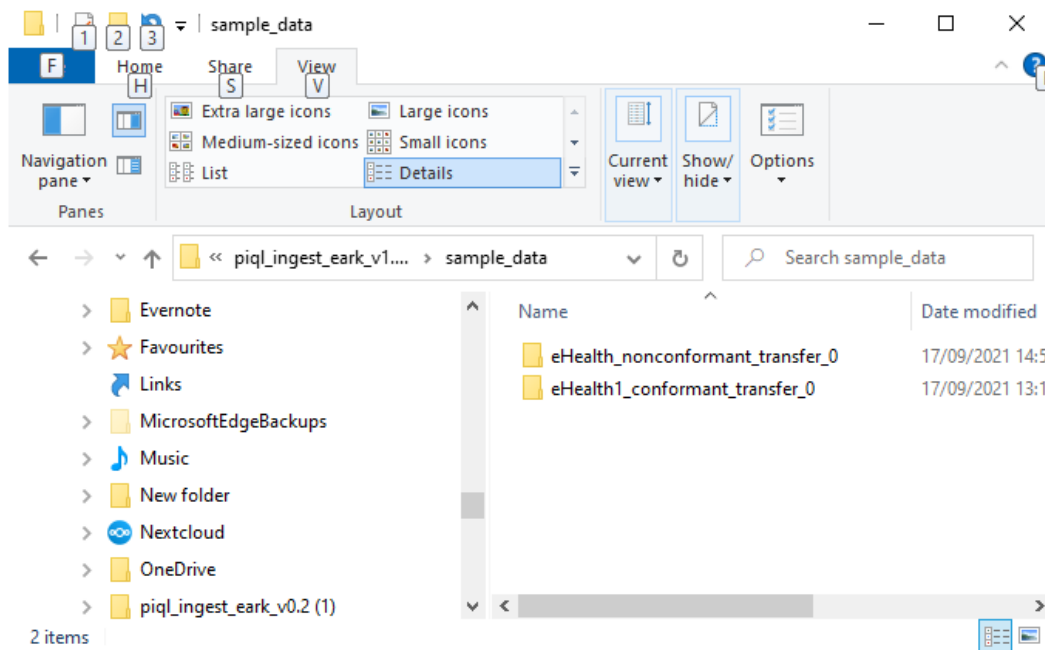


Figure 7 : sample data submissions

Note that you cannot select an entire folder structure using the 'Add Files' button on the application. This can be used to add files once the package itself has been added. Figure 8 shows the SIP Creator window with the eHealth1 conformant package added.

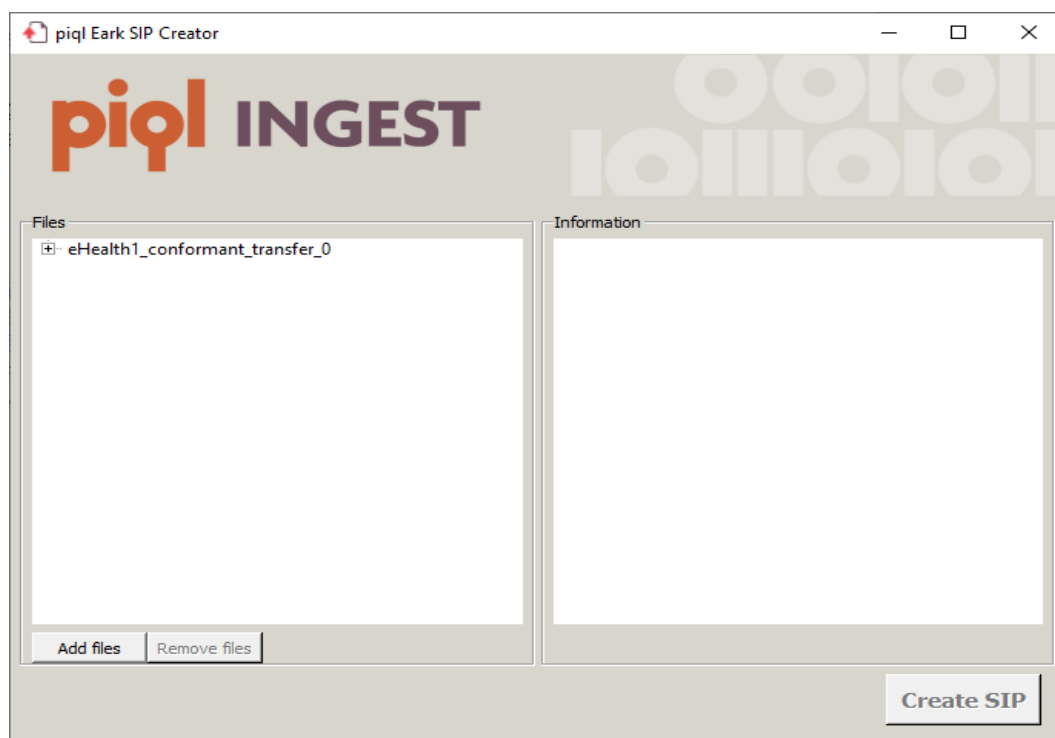
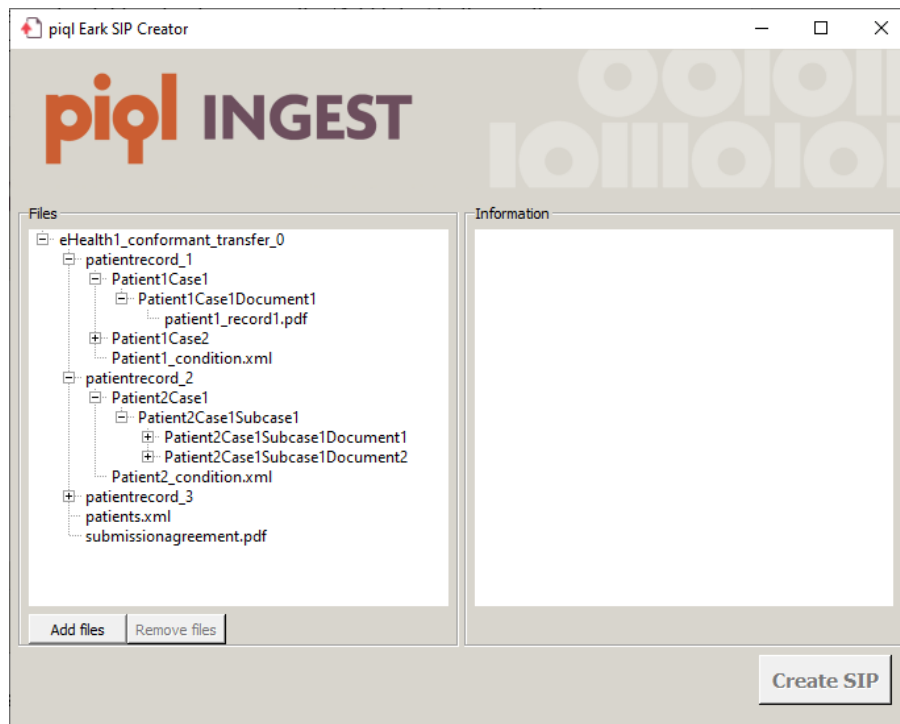


Figure 8 : SIP Creator window with package added



Figure 9 also shows how the package can be inspected to make sure that it is complete.



*Figure 9 : Submission inspection*

Once you are confident that the submission is complete, you can select the package in the inspector window and press 'Create SIP' to start processing.

The tool will now present a metadata editing template with four tabs:

1. Software Version – this is pre-filled and should not be edited.
2. Submission Agreements
  - a. Path or URL to submission agreement – by default this is the \documentation folder of the SIP, but if a submission agreement has not been included in the package, a master reference can be included by means of a URL.
  - b. An identifier or reference code for the submission agreement – default is blank.
  - c. Path or URL to the previous submission agreement – default is blank.
  - d. An identifier or reference code for the previous submission agreement – default is blank.
3. Agents
  - a. Creator organization name – default blank.
  - b. Creator organization identifier – default is blank.
  - c. Archive organization name – default is blank.
  - d. Archive organization identifier – default is blank.

- e. Preservation organization name – default is blank.
- f. Preservation organization identifier – default is blank.
- g. Submitter name (individual) – default is blank.
- h. Submitter details (e.g. email address) – default is blank.

#### 4. Metadata schemas

- a. Patient personal information schema name – default is patients.xsd
- b. Path or URL to patient personal information schema – default is /schemas/patient.xsd
- c. Patient clinical information schema name – default is condition.xsd
- d. Patient clinical information schema location – default is /schémas/condition.xsd

Once the metadata is complete, the 'Create SIP' button can be pressed. The output of the process can be found in the /outputs folder as seen in figure 10.

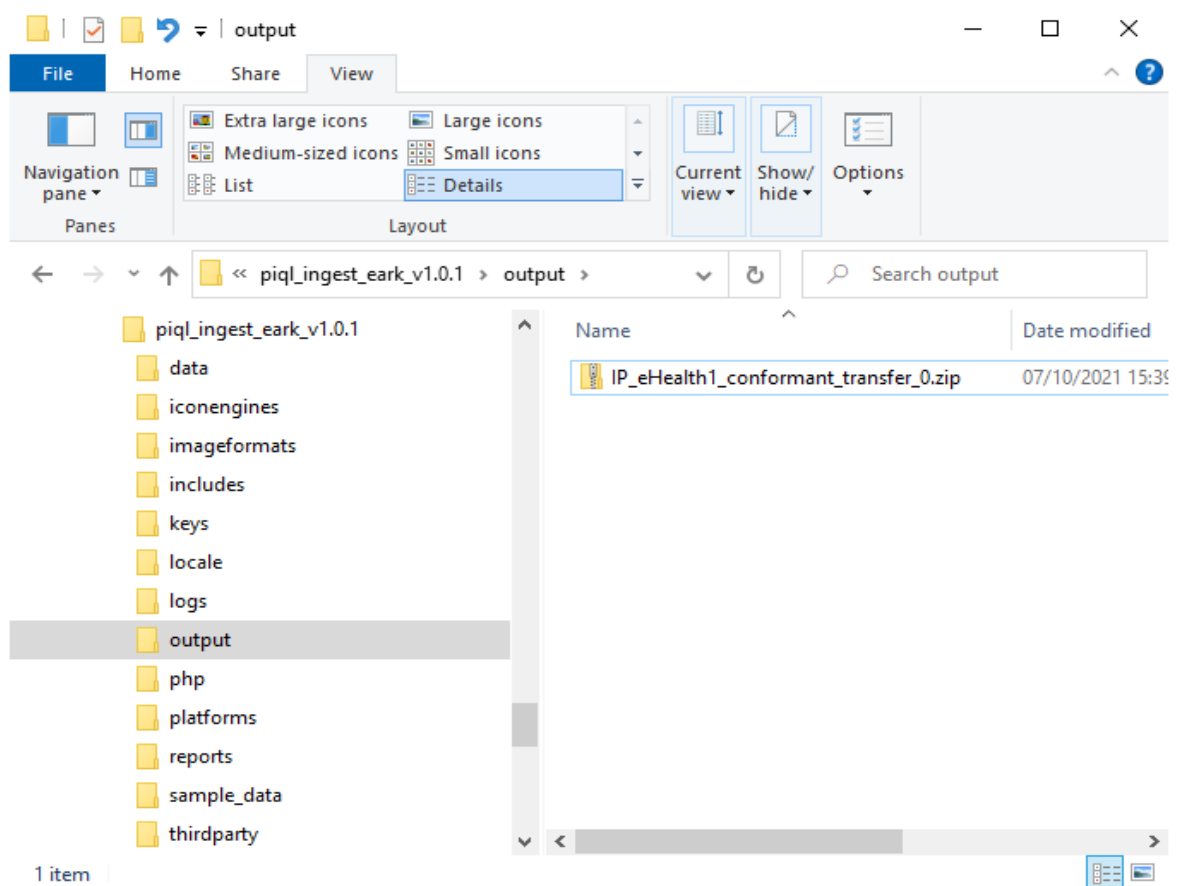


Figure 10 : SIP in output folder

Note that if there is an output SIP in the output folder the same submission cannot be processed, the output SIP must be deleted first.