Test protocol for sending a

MedCom Acknowledgement

21-11-23

The test protocol relates to the following standard:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of the standard ENG** | **Name of the standard DK** | **Version** | **Type** |
| Standard: DK MedCom Acknowledgement | DK MedCom Kvittering | 2.0.0 | HL7 FHIR |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | | | |
| **Version** | **Initials** | **Date** | **Description** |
| 2.0.0 | MBU | 24-05-2023 | First publication in English |
| 2.0.1 | MBU | 30-11-2023 | Clarification about Acknowledgement type AE and AR added. Furthermore, a test step regarding acknowledgement of type [ACK AR] followed by an new acknowledgement has been added |

Content

[1 Introduction 1](#_Toc151468768)

[1.1 Purpose 1](#_Toc151468769)

[1.2 Prerequisites for live test 1](#_Toc151468770)

[1.3 Documentation of self-test 2](#_Toc151468771)

[1.4 Background material 3](#_Toc151468772)

[1.5 Test examples 4](#_Toc151468773)

[1.6 Test tools 5](#_Toc151468774)

[1.7 Test result 5](#_Toc151468775)

[2 Information about vendor, system under test (SUT) and test result information 6](#_Toc151468776)

[2.1 Information about the vendor 6](#_Toc151468777)

[2.2 Information about system under test (SUT) 6](#_Toc151468778)

[2.3 Information about the test result 6](#_Toc151468779)

[3 The test 7](#_Toc151468780)

[3.1 Documentation of the test 8](#_Toc151468781)

[3.2 Test of TouchStone testscripts 9](#_Toc151468782)

[3.3 Test of requirements to content and flow/workflows 10](#_Toc151468783)

[3.4 Test of general technical requirements 14](#_Toc151468784)

# Introduction

This is a test protocol for receiving a MedCom Acknowledgement, hereafter referred to as Acknowledgement.

All documentation concerning MedCom Acknowledgement and Governance (see [Background material](#_Baggrundsmaterialer_1)) will be the subject of testing, and the test protocol will be continuously updated to reflect the requirements in the best way possible.

Versioning of the test protocol will follow the major and minor versions of the standard but may have a patch version that is different from the standard’s patch version.

The system under test (SUT) must be able to receive a MedCom Acknowledgement, (DK: MedCom Kvittering), when a MedCom FHIR message is sent.

## Purpose

The test protocol forms the basis for the tests, which must ensure that SUT complies with the established rules and requirements for the standard. The test protocol also forms the basis for the self-test that vendors carry out prior to a live test.

## Prerequisites for live test

The following prerequisites must be met prior to the live test:

1. The vendor has read the following standard documentation:
   * [Use cases](#_Baggrundsmaterialer_1)
   * [Implementation Guide](#_Baggrundsmaterialer_1)
   * [Governance](#_Baggrundsmaterialer_1)
   * And other relevant materials, cf. the [background material](#_Baggrundsmaterialer_1).
2. The vendor has performed self-test, approved by MedCom.
3. The vendor is using the same version of SUT during self-test and live test.
4. Approval requires that the SUT is approved for sending an Acknowledgement.

## Documentation of self-test

**Self-test**

**Prior to the test, the vendor must have performed self-test, including successfully completed TouchStone self-tests, which are approved by MedCom.**

The self-test is documented by the vendor completing this test protocol.

For self-tests, only the following column must be completed by the vendor:

* [Current result]: is filled in with the results of the self-test and relevant descriptions.

Other columns are reserved for MedCom.

**During the self-test the vendor must document the test results by saving relevant files and screen dumps, and subsequently send these in a combined ZIP file (together with the completed test protocol) to** [**fhir@medcom.dk**](mailto:fhir@medcom.dk)**.**

All files and screen dumps must be named with

* Standard name
* The number of the relevant test step
* Consecutive letter
* File type

Example: Acknowledgement\_3.4\_A.xml

## 

## Background material

| **Name** | **Version[[1]](#footnote-2)** | **Link/reference** | **Description** |
| --- | --- | --- | --- |
| Acknowledgement documentation site | 2.0.X | <https://medcomdk.github.io/dk-medcom-acknowledgement/> | Documentation site with references to all relevant documentation, including:   * Use cases (and matching test scripts) to be used in TouchStone * Technical specifications |
| Acknowledgement Implementation Guide | 2.0.X | <https://medcomfhir.dk/ig/acknowledgement/> | The FHIR technical guidelines for the standard. |
| Governance for MedCom FHIR | 1.0.X | <https://medcomdk.github.io/MedCom-FHIR-Communication/> | Governance for MedCom’s FHIR standards, which describes general rules for all MedCom standards and specific rules for this standard, as well as for sending. |
| SOP 7.2 for MedCom’s test and certification | 2.X | <http://svn.medcom.dk/svn/qms/Offentlig/SOPer/SOP-7.2-MedComs%20test%20og%20certificering_godkendelse.docx> | Description of test and certification of MedCom standards and other tests courses. |

## Test examples

|  |  |  |
| --- | --- | --- |
| **Name** | **Link/reference** | **Description** |
| Test examples | <https://medcomfhir.dk/ig/acknowledgementtestscript/> |  |

## Test tools

|  |  |  |
| --- | --- | --- |
| **Navn** | **Link/reference** | **Description** |
| FHIR server with MedCom profiles | <https://fhir.medcom.dk/> | Public server that validates against MedCom's FHIR profiles. It is permitted to use the server for testing the upload/download of FHIR resources. |
| TouchStone | <https://touchstone.aegis.net/touchstone/> | Test tool for testing the FHIR standard.  The vendor can get access to TouchStone as an organisation - either through a license that MedCom supplies (inquiry at [fhir@medcom.dk](mailto:fhir@medcom.dk)), or a license that the vendor has acquired itself.  Find [instructions for TouchStone](https://medcomdk.github.io/MedComLandingPage/assets/documents/TouchStoneGettingStarted.html) here. |
| TouchStone test scripts | <https://medcomfhir.dk/ig/acknowledgementtestscript/> | Test scripts relevant for the standard.  Find [instructions for TouchStone here](https://medcomdk.github.io/MedComLandingPage/assets/documents/TouchStoneGettingStarted.html). |

## Test result

The result for each test step is categorised based on the table below:

| **Marking** | **F1** | **F2** | **F3** | **F4** | **Ok** | **Not relevant** |
| --- | --- | --- | --- | --- | --- | --- |
| **Evaluation** | **Critical** | **Serious** | **Significant** | **Less significant** | **Approved** | **Not an error** |

To get the test and certification approved, the test protocol must consist exclusively of [F4] as well as [OK] results. All [F1], [F2] and [F3] must, therefore, be fixed prior to final approval.

Approval requires that SUT is approved for receiving a FHIR acknowledgement (DK: Kvittering)

For further information, please read: [MedCom’s test and certification.](#_Background_material_1)

# Information about vendor, system under test (SUT) and test result information

## Information about the vendor

This table must be completed by **the vendor** prior to the test.

|  |  |
| --- | --- |
| Company | Completed by vendor |
| Address | Completed by vendor |
| Contact person | Completed by vendor |
| Telephone | Completed by vendor |
| E-mail | Completed by vendor |

## Information about system under test (SUT)

This table must be completed by **the vendor** prior to the test.

|  |  |
| --- | --- |
| System | Completed by vendor |
| Version | Completed by vendor |
| Description | Completed by vendor |
| Test type | Self-test  Final test/certification |

## Information about the test result

Note: This table must be completed by MedCom when the test has been completed.

|  |  |
| --- | --- |
| Test date | 2023-01-01 |
| Test location |  |
| Approved | Yes  No |
| Remarks | Completed by MedCom |
| Carried out by | The name of the MedCom responsible (initials) for this test |

# The test

This section describes the requirements which SUT must meet before final approval.

The test is divided into three sections:

1. Test of TouchStone testscripts
2. Test of requirements for content and flow/workflows
3. Test of technical requirements

Test participants will be asked to complete tests as described in the tables.

## Documentation of the test

**Documentation of the test**

As valid documentation, the test participant or the test manager must document completion by continuous screen dumps (.png/.jpeg) and/or files/log files (.xml/.json). **Before the test, it is agreed on who is responsible for this.**

The following applies:

* The files must be viewable in a standard tool and must not require further processing by MedCom
* All files and screen dumps must be named with:
  + The name of the standard
  + The number of the relevant test setup
  + Consecutive letter
  + File type

*Examples: HospitalNotfication\_3.4\_A.xml, HospitalNotification\_3.4\_B.xml*

If the vendor has documented the test themselves, the files must be sent in a ZIP file to [fhir@medcom.dk](mailto:fhir@medcom.dk).

## Test of TouchStone testscripts

The purpose of these tests is to ensure that SUT generates the message technically correct and complies with the rules in the [Implementation Guide](#_Background_material_1).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
|  | **Optional test step:**  Run all test scripts for use cases and user flows in TouchStone. |  | All test scripts completed without errors. |  | Choose |

## Test of requirements to content and flow/workflows

The purpose of these tests is to ensure that the standard is implemented with satisfactory quality, i.e. that implementation meets the business requirements for flow and content as described in the [use case material](#_Background_material_1). These test steps are mainly for the user interface.

The table below lists the use cases which are tested in relation to content and flow/workflows. The table also shows the direct references to the use cases in the [use case material](#_Background_material_1).

|  |  |  |  |
| --- | --- | --- | --- |
| [**Use case**](#_Background_material_1) | **Description** | **Section** | **Example file** |
| R.TC2 | Create an Acknowledgement   * Positive Acknowledgement [ACK AA] * Negative Acknowledgement [ACK AE] * Negative Acknowledgement [ACK AR] | 3.3.1 | <https://medcomfhir.dk/ig/acknowledgementtestscripts/> |
| R.TC3 | Send Acknowledgement | 3.3.1 | <https://medcomfhir.dk/ig/acknowledgementtestscripts/> |

### R.TC2 Create Acknowledgement and R.TC3 Send Acknowledgement

| **Test step #** | **Action** | **Test data** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | **Send Acknowledgement of the type [ACK AA]**  Load the test data file with a valid MedCom FHIR message. |  | The MedCom FHIR message was validated positively and, thus, is loaded correctly in SUT. |  | Choose |
|  | Demonstrate that the system has created an Acknowledgement of the type [ACK AA] on the basis of a positive validation of a MedCom FHIR message. |  | A positive Acknowledgement of the type [ACK AA] has been created. |  | Choose |
|  | Demonstrate that the system has sent an Acknowledgement of the type [ACK AA]. |  | Acknowledgement of the type [ACK AA] has been sent correctly. |  | Choose |
|  | **Send Acknowledgement of the type [ACK AE] due to invalid content of the original message**  Load the test data file with an invalid MedCom FHIR message. |  |  |  | Choose |
|  | Demonstrate that the system has created a negative Acknowledgement on the basis of a negative validation of a MedCom FHIR message. |  | A negative Acknowledgement of the type [ACK AE] is shown. |  | Choose |
|  | Demonstrate that the system has sent an Acknowledgement of the type [ACK AE]. |  | Acknowledgement of the type [ACK AE] has been sent correctly. |  | Choose |
|  | **Negative Acknowledgement [ACK AR] due to technical error in the system.**  Explain how SUT handles Acknowledgement of the type [ACK AR]. If possible, load a valid MedCom FHIR message to demonstrate how this is handled. |  | An explanation has been given of how SUT handles a negative Acknowledgement of the type [ACK AR] and/or SUT has generated an Acknowledgement of the type [ACK AR]. |  | Choose |
|  | If an Acknowledgement of the type [ACK AR] has been generated in test step 3.3.1.7, demonstrate then that SUT has sent this Acknowledgement. |  | Acknowledgement of the type [ACK AR] has been sent correctly. |  | Choose |
|  | If the technical error from test step 3.3.1.8 is resolved and the message can be loaded in, demonstrate that the SUT generates an Acknowledgement based on the validation.  If the validation of the loaded MedCom message is positive then the Acknowledgement shall be [ACK AA]  If the validation of the loaded message is negative then the Acknowledgement shall be [ACK AE] |  | An Acknowledgement has been created and sent. |  |  |

## Test of general technical requirements

The purpose of these test steps is to ensure that the technical sending of a DK MedCom Acknowledgement is implemented with satisfactory quality, i.e., meets the governance for message communication on a general level as well as governance for DK MedCom Acknowledgement as described in section 1.4 Background material.

| **Test step #** | **Action** | **Test data** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | **Handling of duplicate**  Reload the test data file with the received valid MedCom FHIR message from test step 3.3.1.1.  If a message is considered a duplicate, explain or demonstrate how SUT handles this.  Show that SUT sends an Acknowledgement with the same content which SUT returned first time SUT has validated the FHIR message.  For example:  If the first Acknowledgement was of the type [ACK AA] then Acknowledgement of the type [ACK AA] must be resent.  In addition, demonstrate that Bundle.id is changed if the Acknowledgement is resent. |  | SUT sends Acknowledgement with the corresponding to the Acknowledgement returned when the first message was received.  It is also demonstrated that the Bundle.id is changed when the Acknowledgement is resent. |  | Choose |
|  | **Time stamps**  Generate a random Acknowledgement (for example an Acknowledgement of the type [ACK AA]) at 10 am and demonstrate that the Bundle.timestamp shows the time of the creation of the Acknowledgement.  Demonstrate that Provenance.occuredDateTime[x] and Provenance.recorded show the time of sending and not the time of creation. |  | Bundle.timestamp = 10:00  Provenance.occuredDateTime[x] =10:10  Provenance.recorded=10:10 |  | Choose |
|  | Explain or demonstrate that Bundle.timestamp, Provenance.occuredDateTime[x] and Provenance.recorded are changed if the Acknowledgement is resent. |  | Bundle.timestamp, Provenance.occuredDateTime[x] and Provenance.recorded are changed. |  | Choose |
|  | **Correct embedment of messages in VANSEnvelope**  Select a random Acknowledgement and demonstrate that the Acknowledgement is embedded correctly in a VANSEnvelope and contains a postfix with the Acknowledgement type in the name element. |  | The message contains a valid Acknowledgement and is embedded correctly in a VANSEnvelope.  The VANSEnvelope contains:   * Format * Name (incl. postfix with the Acknowledgement type) * Version |  | Choose |
|  | **Overview of Acknowledgement messages**  Demonstrate that SUT has an overview of the sent Acknowledgement messages where it is also evident which MedCom FHIR message is associated with the Acknowledgement. |  | An overview of sent Acknowledgement messages and which MedCom FHIR messages they are associated to, is shown. |  | Choose |

1. X expresses versioning on patch level, including minor corrections which are backward compatible. [↑](#footnote-ref-2)