Community Medication Prescription and Dispense (CMPD)

Trial Implementation

Date: October 23, 2015
Author: IHE Pharmacy Technical Committee
Email: pharmacy@ihe.net

Please verify you have the most recent version of this document. See here for Trial Implementation and Final Text versions and here for Public Comment versions.
Foreword

This is a supplement to the forthcoming IHE Pharmacy Technical Framework. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

This supplement is published on October 23, 2015 for trial implementation and may be available for testing at subsequent IHE Connectathons. The supplement may be amended based on the results of testing. Following successful testing it will be incorporated into the forthcoming Pharmacy Technical Framework. Comments are invited and may be submitted at http://www.ihe.net/Pharmacy_Public_Comments.

This supplement describes changes to the existing technical framework documents.

“Boxed” instructions like the sample below indicate to the Volume Editor how to integrate the relevant section(s) into the relevant Technical Framework volume.

Amend Section X.X by the following:

Where the amendment adds text, make the added text **bold underline**. Where the amendment removes text, make the removed text **bold strikethrough**. When entire new sections are added, introduce with editor’s instructions to “add new text” or similar, which for readability are not bolded or underlined.

General information about IHE can be found at: http://www.ihe.net.

Information about the IHE Pharmacy domain can be found at: http://ihe.net/IHE_Domains.

Information about the organization of IHE Technical Frameworks and Supplements and the process used to create them can be found at: http://ihe.net/IHE_Process and http://ihe.net/Profiles.

The current version of the IHE Pharmacy Technical Framework can be found at: http://ihe.net/Technical_Frameworks.
CONTENTS

Introduction ..................................................................................................................................... 6
Open Issues and Questions ........................................................................................................ 7
Closed Issues .............................................................................................................................. 7

Volume 1 – Integration Profiles ................................................................................................... 9
1.n Copyright Permission ........................................................................................................... 9
2.1 Dependencies among Integration Profiles ........................................................................... 9
2.2.4 Community Medication Prescription and Dispense Integration Profile ...................... 9
4 Community Medication Prescription and Dispense Integration Profile .............................. 11

4.1 Actors/Transactions ........................................................................................................... 12
4.1.1 Actors ............................................................................................................................. 15
4.1.1.1 Community Pharmacy Manager ......................................................................... 15
4.1.1.2 Medication Treatment Planner ........................................................................... 15
4.1.1.3 Prescription Placer .............................................................................................. 15
4.1.1.4 Pharmaceutical Adviser ...................................................................................... 15
4.1.1.5 Medication Dispenser ......................................................................................... 16
4.1.1.6 Repository actors .................................................................................................. 16
4.1.2 Transactions .................................................................................................................. 17
4.1.2.1 Query Pharmacy Documents .............................................................................. 17
4.1.2.2 Registry Stored Query ........................................................................................ 18
4.1.2.3 Provide and Register Document Set -b ............................................................... 19
4.1.2.4 Retrieve Document Set ....................................................................................... 19

4.2 CMPD Integration Profile Options .................................................................................... 19
4.2.1 Provision of Medication List Option .......................................................................... 20
4.2.2 Persistence of Retrieved Documents Option .............................................................. 20
4.2.3 Medication Treatment Planning Option ..................................................................... 20
4.2.4 Workflow Management Option ................................................................................. 20

4.3 CMPD Actor Groupings and Profile Interactions .............................................................. 20

4.4 CMPD Process Flow .......................................................................................................... 21
4.4.1 Use Case community pharmacy-active substance, publish & pull (Scenario 1: “Including validation step”) .............................................................. 22
4.4.1.1 Story Board ......................................................................................................... 24
4.4.1.2 Sequence Diagram ............................................................................................. 24
4.4.2 Use Case community pharmacy-active substance, plan, publish & pull (Scenario 2: “Not including validation step”) .............................................................. 26
4.4.2.1 Story Board ......................................................................................................... 27
4.4.2.2 Sequence Diagram ............................................................................................. 28
4.4.3 Use Case: Physician requests Medication List ........................................................... 30
4.4.3.1 Story Board ......................................................................................................... 30
4.4.3.2 Sequence Diagram ............................................................................................. 31
4.4.4 Use Case: Physician changes/cancels or suspends an unfilled prescription .......... 32
Introduction

The Community Medication Prescription and Dispense Integration Profile (CMPD) describes the process of prescription, validation and dispense of medication in the community domain. This document is a detailed description of the generic implementation structure defined in the Common Parts document¹.

In general, the medication business process consists of five distinct processes, which have to be connected through interactions that transfer information and/or guide the workflow. The following figure shows this flow:

In the Community Pharmacy domain, the process of “administration of medication” can usually not be governed by IT based systems so just the processes “Planning”, “Prescription”, “Pharmaceutical Advice” and “Dispense” are covered by the Community Pharmacy Prescription and Dispense Profile only.

The CMPD Profile is intended to be used in the context of the Pharmacy Content Profiles²:

- Medication Treatment Plan (MTP)

¹ This document is part of the IHE Pharmacy domain and can be obtained from the IHE web site.
² These supplements are part of the IHE Pharmacy domain and can be obtained from the IHE web site.
• Pharmacy Prescription Supplement (PRE)
• Pharmacy Pharmaceutical Advice Supplement (PADV)
• Pharmacy Dispense Supplement (DIS)
• Pharmacy Medication List (PML)

These Content Profiles are based on the Patient Care Coordination (PCC) Technical Framework and define the semantic of the payload transported by the CMPD Profile.

This supplement also references other documents\(^3\). The reader should have already read and understood these documents:

1. [PHARM Common parts document](http://ihe.net/Technical_Frameworks/)
2. [IT Infrastructure Technical Framework Volume 1](http://ihe.net/Technical_Frameworks/)
3. [IT Infrastructure Technical Framework Volume 2](http://ihe.net/Technical_Frameworks/)
4. [IT Infrastructure Technical Framework Volume 3](http://ihe.net/Technical_Frameworks/)
5. HL7® and other standards documents referenced in this document

### Open Issues and Questions

- Grouping of XDW with the Community Pharmacy Manager: What, if the client-side actors (Prescription Placer, Pharmaceutical Adviser, Medication Dispenser) are not allowed to manage the workflow and this should be done by the CPM.

### Closed Issues

- Question: Should be medication processes message or document-based? Decision at F2F meeting in Bordeaux (15./16.04.2010): Community domain is document based with XDS as persistence layer, Hospital domain will be message-based.

- Clarification to whitepaper: In community domain, the term “repository” in the whitepaper is intended to be interpreted as a technical system for persisting documents implementing XDS transactions as interface. XDS registry/repository systems as well as database or other persisting systems are likely to be used for this purpose.

- Changes to whitepaper:

“Consumer” actors will be removed, because they are just relaying transactions (don’t implement any own transactions). Sequence diagrams have been adapted.

---

\(^3\) The first four documents can be located on the IHE Website at [http://ihe.net/Technical_Frameworks/](http://ihe.net/Technical_Frameworks/). The remaining documents can be obtained from their respective publishers.
The transient aspects of “Ordering” are excluded in the profile. This should be generally discussed together with ITI in conjunction with all other “Ordering/Workflow” topics (e.g., Lab, Referral, etc.). (see CP-PHARM-018_v5)

The profile does not yet include the process step of getting the “current medication”, which is needed for checking interactions (ICAs) to the prescribed item.

This issue is closed since the introduction of PML.
Volume 1 – Integration Profiles

1.n Copyright Permission

Add the following to Sections 1.n:

Health Level Seven, Inc. has granted permission to the IHE to reproduce tables from the HL7® standard. The HL7® tables in this document are copyrighted by Health Level Seven, Inc. All rights reserved. Material drawn from these documents is credited where used.

2.1 Dependencies among Integration Profiles

Add the following to Table 2-1:

| Community Pharmacy Prescription and Dispense CMPD | XDS | CMPD Actors are based on XDS Document Source, Document Consumer, Registry and Repository actors and use XDS transactions. | Required to manage query, submission and retrieve of documents. |
| Community Pharmacy Prescription and Dispense CMPD | On-Demand Documents | CMPD Community Pharmacy Manager acts as an On-Demand Document Source Actor | Required to manage request of the Medication List |
| Community Pharmacy Prescription and Dispense CMPD | ATNA | Each CMPD Actor shall be grouped with Secure Node or Secure Application Actor | Required due to XDS grouping. |
| Community Pharmacy Prescription and Dispense CMPD | CT | Each CMPD Actor shall be grouped with the Time Client Actor | Required due to ATNA grouping. |
| Community Pharmacy Prescription and Dispense CMPD | XDW | Some CMPD actors can be optionally grouped with XDW Workflow Management | Optional due to XDW grouping |

Add the following section to Section 2.2:

2.2.4 Community Medication Prescription and Dispense Integration Profile

The Community Medication Prescription and Dispense Integration Profile (CMPD) describes the process of planning, prescription, validation and dispense of medication in the community domain.

The CMPD Profile is intended to be used in the context of the Pharmacy Content Profiles⁴:

⁴ These supplements are part of the IHE Pharmacy domain and can be obtained from the IHE web site.
• Medication Treatment Plan Supplement (MTP)
• Pharmacy Prescription Supplement (PRE)
• Pharmacy Pharmaceutical Advice Supplement (PADV)
• Pharmacy Dispense Supplement (DIS)
• Pharmacy Medication List (PML)

These Content Profiles are based on the Patient Care Coordination (PCC) Technical Framework and define the semantic of the payload transported by the CMPD Profile.

Add Section 4
4 Community Medication Prescription and Dispense Integration Profile

The Community Medication Prescription and Dispense Integration Profile (CMPD) describes the process of planning, prescription, validation and dispense of medication in the community domain.

In general, the medication business process consists of five distinct processes, which have to be connected through interactions that transfer information and/or guide the workflow. The following figure shows this flow:

Figure 4-1: Medication Planning, Prescription and Dispense Process

In the Community Pharmacy domain, the process of “administration of medication” can usually not be governed by IT based systems so just the processes “Plan”, “Prescription”, “Pharmaceutical Advice” and “Dispense” are covered by the Community Pharmacy Prescription and Dispense Profile only.

The CMPD Profile is intended to be used in the context of the Pharmacy Content Profiles:

5 These supplements are part of the IHE Pharmacy domain and can be obtained from the IHE web site.
• Medication Treatment Plan (MTP)
• Pharmacy Prescription Supplement (PRE)
• Pharmacy Pharmaceutical Advice Supplement (PADV)
• Pharmacy Dispense Supplement (DIS)
• Pharmacy Medication List (PML)

These Content Profiles are based on the Patient Care Coordination (PCC) Technical Framework and define the semantic of the payload transported by the CMPD Profile.

4.1 Actors/ Transactions

Figure 4.1-1 shows the actors directly involved in the Community Medication Prescription and Dispense Integration Profile and the relevant transactions between them. Other actors that may be indirectly involved due to their participation in the XDS integration profiles, etc., are not necessarily shown.

**XDS Transactions defined as**
- Registry Stored Query [ITI-18]
- Provide and Register Document Set-a [ITI-41]
- Retrieve Document [ITI-43]

**Important notes:**
1) Only for retrieving the Medication List, if „Provision of Medication List” option is used at Community Pharmacy Manager
2) If „Persistence of Retrieved Documents” option is used at Community Pharmacy Manager

**Figure 4.1-1: Community Medication Prescription and Dispense Actor Diagram**
Important Note:

The Community Pharmacy Manager Actor (CPM) is currently restricted to perform data “filtering” by the PHARM-1 “Query Pharmacy Documents” transaction as well as data “relaying” of the ITI-43 “Retrieve Document Set” transaction. All other XDS transactions are performed directly between the client actors (Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser, Medication Dispenser) and the Registry/Repository actors (Medication Treatment Plan, Prescription, Pharmaceutical Advice, Dispense repositories).

The “relaying” of transactions for client actors is not shown in this Actor diagram for readability. Please see chapter “CMPD Implementation scenarios” for details to the usage of the “relaying” functionality of the CPM in case e.g., of a multi-domain implementation scenario.

Table 4.1-1 lists the transactions for each actor directly involved in the Community Medication Prescription and Dispense Profile. In order to claim support of this Integration Profile, an implementation must perform the required transactions (labeled “R”). Transactions labeled “O” are optional. A complete list of options defined by this Integration Profile and that implementations may choose to support is listed in Volume 1, Section 4.2.

<table>
<thead>
<tr>
<th>Actors</th>
<th>Transactions</th>
<th>Optionality</th>
<th>Section in Vol. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community Pharmacy Manager</td>
<td>Registry Stored Query (ITI-18)</td>
<td>R</td>
<td>ITI-TF-2a:3.18</td>
</tr>
<tr>
<td></td>
<td>(acting as a Document Consumer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retrieve Document Set (ITI-43)</td>
<td>R</td>
<td>ITI-TF-2b:3.43</td>
</tr>
<tr>
<td></td>
<td>(acting as a Document Consumer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Register On-Demand Document Entry (ITI-61)</td>
<td>O^6</td>
<td>ITI-TF-2b:3.61</td>
</tr>
<tr>
<td></td>
<td>(acting as an On-Demand Document Source)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Query Pharmacy Documents (PHARM-1)</td>
<td>R</td>
<td>PHARM-TF-2:3.1</td>
</tr>
<tr>
<td>Medication Treatment Planner</td>
<td>Registry Stored Query (ITI-18)</td>
<td>O^7</td>
<td>ITI-TF-2a:3.18</td>
</tr>
<tr>
<td></td>
<td>Provide and Register Document Set-b (ITI-41)</td>
<td>R</td>
<td>ITI-TF-2b:3.41</td>
</tr>
<tr>
<td></td>
<td>Retrieve Document Set (ITI-43)</td>
<td>R</td>
<td>ITI-TF-2b:3.43</td>
</tr>
</tbody>
</table>

^6 R if „Provision of Medication List“ Option is used

^7 R if „Workflow Management“ Option is used
<table>
<thead>
<tr>
<th>Actors</th>
<th>Transactions</th>
<th>Optionality</th>
<th>Section in Vol. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prescription Placer</td>
<td>Query Pharmacy Documents (PHARM-1)</td>
<td>R(^8)</td>
<td>PHARM-TF-2:3.1</td>
</tr>
<tr>
<td></td>
<td>Registry Stored Query (ITI-18)</td>
<td>O(^9)</td>
<td>ITI-TF-2a:3.18</td>
</tr>
<tr>
<td></td>
<td>Provide and Register Document Set-b (ITI-41)</td>
<td>R</td>
<td>ITI-TF-2b:3.41</td>
</tr>
<tr>
<td></td>
<td>Retrieve Document Set (ITI-43)</td>
<td>R</td>
<td>ITI-TF-2b:3.43</td>
</tr>
<tr>
<td></td>
<td>Query Pharmacy Documents (PHARM-1)</td>
<td>R(^10)</td>
<td>PHARM-TF-2:3.1</td>
</tr>
<tr>
<td>Pharmaceutical Adviser</td>
<td>Registry Stored Query (ITI-18)</td>
<td>O(^11)</td>
<td>ITI-TF-2a:3.18</td>
</tr>
<tr>
<td></td>
<td>Provide and Register Document Set-b (ITI-41)</td>
<td>R</td>
<td>ITI-TF-2b:3.41</td>
</tr>
<tr>
<td></td>
<td>Retrieve Document Set (ITI-43)</td>
<td>R</td>
<td>ITI-TF-2b:3.43</td>
</tr>
<tr>
<td></td>
<td>Query Pharmacy Documents (PHARM-1)</td>
<td>R(^12)</td>
<td>PHARM-TF-2:3.1</td>
</tr>
<tr>
<td>Medication Dispenser</td>
<td>Registry Stored Query (ITI-18)</td>
<td>O(^13)</td>
<td>ITI-TF-2a:3.18</td>
</tr>
<tr>
<td></td>
<td>Provide and Register Document Set-b (ITI-41)</td>
<td>R</td>
<td>ITI-TF-2b:3.41</td>
</tr>
<tr>
<td></td>
<td>Retrieve Document Set (ITI-43)</td>
<td>R</td>
<td>ITI-TF-2b:3.43</td>
</tr>
<tr>
<td></td>
<td>Query Pharmacy Documents (PHARM-1)</td>
<td>R(^14)</td>
<td>PHARM-TF-2:3.1</td>
</tr>
<tr>
<td>Repository actors</td>
<td>Registry Stored Query (ITI-18)</td>
<td>R</td>
<td>ITI-TF-2a:3.18</td>
</tr>
<tr>
<td>Medication Treatment Plan(^15)</td>
<td>Provide and Register Document Set-b (ITI-41)</td>
<td>R</td>
<td>ITI-TF-2b:3.41</td>
</tr>
<tr>
<td>Prescription</td>
<td>Retrieve Document Set (ITI-43)</td>
<td>R</td>
<td>ITI-TF-2b:3.43</td>
</tr>
<tr>
<td>Pharmaceutical Advice</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dispense</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medication List(^16)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8 O if „Workflow Management“ Option is used  
9 R if „Workflow Management“ Option is used  
10 O if „Workflow Management“ Option is used  
11 R if „Workflow Management“ Option is used  
12 O if „Workflow Management“ Option is used  
13 R if „Workflow Management“ Option is used  
14 O if „Workflow Management“ Option is used  
15 If „Medication Treatment Planning Option“ is used  
16 If „Persistence of Retrieved Documents“ Option is used at Community Pharmacy Manager
4.1.1 Actors

4.1.1.1 Community Pharmacy Manager

The main role of this actor consists in providing the business logic for status management and other purposes. It may also serve as provider of the business logic for creating the Medication List if “Provision of Medication List” Option is supported. As a second role it acts as a “relaying role” where certain standard XDS communication is routed through for providing the possibility of applying project-specific business logic on it.

It provides special query-transactions which consuming actors (Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser) use for reducing the amount of data flowing to them. They return just “relevant” information for specific purposes (e.g., returning just all “active” prescriptions ready for being validated or dispensed together with all related documents).

Furthermore it may provide special query-transactions which consuming actors (Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser) use to request a Medication List to a patient. Fulfilling the request the actor gathers and assembles Medication Treatment Plan-, Prescription- and Dispense items to a Medication List document according to the “Pharmacy Medication List” (PML) Profile. Subsequently this resulting document is returned to the requesting actor. For this functionality this actor acts as an ITI On-Demand Document Source Actor as described in the “On-Demand documents” supplement.

This actor is usually a system actor without human participation.

4.1.1.2 Medication Treatment Planner

The main role of this actor consists in adding a new Medication Treatment Plan Item. It sends the cancelation of the planned item or its discontinuation, as well. In order to fulfill this task, the Medication Treatment Plan Planner retrieves the current set of planned medications of the patient.

4.1.1.3 Prescription Placer

The main role of this actor consists in placing the prescription (initial or modified in case of a substitution of invalidation, for example). It sends the cancelation of the prescription or its discontinuation, as well. In order to fulfill this task, the Prescription Placer retrieves the current treatment of the patient and medication already dispensed recently.

4.1.1.4 Pharmaceutical Adviser

This actor is responsible for the validation of prescriptions from a pharmacist’s perspective. Therefore, it receives the initial prescription, validates it and sends it back (accepted, cancelled, modified, substitution of pharmaceutical product); therefore it provides the pharmaceutical advice. To perform this task it checks the current treatment.
Pharmaceutical Advisers (e.g., automated ICA check modules) may also provide “draft” advices which don’t affect the status of a prescription but serve as a foundation for the advice performed by another Pharmaceutical Adviser.

### 4.1.1.5 Medication Dispenser

This actor is responsible for the process of dispensing medication to the patient, e.g., fulfilling the prescription. Therefore it produces the information on the medication dispensed to the patient. In order to achieve this, it receives prescriptions already validated. It also confirms drug availability for administration and it receives the administration plan and administration reports. This actor may be implemented as the point of sale software of a community pharmacy or the hospital pharmacy module of a hospital information system. The human actor behind this system actor is usually a pharmacist or a pharmacist assistant.

### 4.1.1.6 Repository actors

Formally the Community Pharmacy process defines different “repositories” for Medication Treatment Plans, Prescriptions, Pharmaceutical Advices and Dispenses, but they shall be seen as abstract repository-roles for persisting the appropriate document types the documents, not as XDS repositories defined in the “Cross Document Sharing” (XDS) Integration Profile of the ITI Technical Framework.

This profile rather makes use of the XDS Profile for defining abstract XDS registry and repository actors for modeling the abstract repository-roles for real implementations.

Description of the abstract repository-roles:

- **Medication Treatment Plan Repository**
  
  This repository contains the medication added to the patient’s plan from the Medication Treatment Planner and may receive updates to the current planning (cancelations, changes, etc.). It also provides information about the current planned medication to other actors such as the Community Pharmacy Manager.

- **Prescription Repository**
  
  This repository contains the medication prescribed to the patient from the Prescription Placer and may receive updates to the current treatment (cancelations, changes, etc.). It also provides information about the current prescribed medication to other actors such as the Community Pharmacy Manager.

- **Pharmaceutical Advice Repository**
  
  This repository contains the pharmaceutical advice issued by the Pharmaceutical Adviser (typically a pharmacist). It provides this information to other actors such as the Community Pharmacy Manager.

- **Dispensed Medication Repository**

Rev. 1.6 – 2015-10-23  Copyright © 2015: IHE International, Inc.
• This repository contains the medication actually dispensed to the patient; this information is received from the Medication Dispenser. The Dispensed Medication Repository provides the medication record of the patient to other actors such as the Community Pharmacy Manager.

Conforming to the ITI XDS Technical Framework, registry actors are used for storing metadata of the submitted documents, the repository actors store the actual documents.

Implementation scenarios in real-world projects will most likely differ from the topology of having exactly three repositories. They may vary from single XDS affinity domain scenarios with just one registry/repository system for storing all document-types to most complex scenarios including many different XDS affinity domains for covering the organizational and strategic need of separation of the participating parties (e.g., Prescribers and Pharmacists).

All mechanisms defined in the XDS Integration Profile for accessing XDS Registry/Repository systems apply and may be used for communicating, e.g., “Cross Community Access” (XCA).

4.1.2 Transactions

4.1.2.1 Query Pharmacy Documents

This transaction defines how a querying actor has to query the Community Pharmacy Manager for prescriptions (PRE) and dispenses (DIS) and their related documents. Related documents are Pharmaceutical Advice (PADV) and Dispense (DIS) documents (just in case of querying prescriptions).

Querying actors may be:

• Medication Treatment Planner
• Prescription Placer
• Pharmaceutical Adviser
• Medication Dispenser

This transaction provides a set of specialized queries:

(1) Specialized queries allow the finding of prescriptions and their related documents for specific purposes (e.g., for validation).

These are:

• FindPrescriptions
  Find prescriptions and their related documents
• FindDispenses
  Find dispense documents and their related documents
• FindMedicationTreatmentPlans
Find planned medications and their related documents

- **FindPrescriptionsForValidation**
  
  Find prescriptions and their related documents containing Prescription Items ready to be validated

- **FindPrescriptionsForDispense**
  
  Find prescriptions and their related documents containing Prescription Items ready to be dispensed

Both specialized queries can be parameterized to …

1. … either check the status of a given prescription (e.g., if the patient shows the printed prescription to the operator and the prescription ID can be read off it).
   
   In this case the ID of the given prescription is set in the query parameters - if the prescription is in the requested status (e.g., “ready for dispense”) it shows up in the query result (together with its related documents); otherwise the query result is empty which indicates that the given prescription is not in the requested status.

2. … or to search for prescriptions which are in a specific status (e.g., if the patient has no printed prescription and the implementation allows searching for prescriptions).
   
   In this case the query returns all prescriptions which are in the requested status (e.g., “ready for dispense”). The operator can choose and pick the right one.

(2) A query for requesting the Medication List, if “Provision of Medication List” Option is supported.

This is:

- **FindMedicationList**
  
  Find the Medication List to a patient.

**4.1.2.2 Registry Stored Query**

This transaction is used by a Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser Actor to a registry actor (Medication Treatment Plan/Prescription/Pharmaceutical Advice/Dispensed medication registry) in order to query for Medication Treatment Plan, Prescription, Pharmaceutical Advice or Dispense documents based on the querying actor’s query parameters.

See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2a:3.18).
4.1.2.3 Provide and Register Document Set-b

This transaction is sent by a Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser Actor to a repository actor (Medication Treatment Plan/Prescription/Pharmaceutical Advice/Dispensed Medication Repository) in order to submitting one or more Medication Treatment Plan, Prescription, Pharmaceutical Advice or Dispense documents. See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2b:3.41).

4.1.2.4 Retrieve Document Set

This transaction is sent by a Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser Actor to a repository actor (Medication Treatment Plan/Prescription/Pharmaceutical Advice/Dispensed Medication Repository) or the Community Pharmacy Manager Actor in order to retrieve one or more Medication Treatment Plan, Prescription, Pharmaceutical Advice or Dispense documents.

See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2b:3.43).

4.2 CMPD Integration Profile Options

Options that may be selected for this Integration Profile are listed in the Table 4.2-1 along with the Actors to which they apply. Dependencies between options when applicable are specified in notes.

<table>
<thead>
<tr>
<th>Actor</th>
<th>Options</th>
<th>Vol. &amp; Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community Pharmacy Manager</td>
<td>Provision of Medication List</td>
<td>PHARM TF-1: 4.2.1</td>
</tr>
<tr>
<td></td>
<td>Persistence of Retrieved Documents</td>
<td>PHARM TF-1: 4.2.2</td>
</tr>
<tr>
<td></td>
<td>Medication Treatment Planning</td>
<td>PHARM TF-1: 4.2.3</td>
</tr>
<tr>
<td>Medication Treatment Planner</td>
<td>Workflow Management</td>
<td>PHARM TF-1: 4.2.4</td>
</tr>
<tr>
<td></td>
<td>Provision of Medication List</td>
<td>PHARM TF-1: 4.2.1</td>
</tr>
<tr>
<td>Prescription Placer</td>
<td>Workflow Management</td>
<td>PHARM TF-1: 4.2.4</td>
</tr>
<tr>
<td></td>
<td>Provision of Medication List</td>
<td>PHARM TF-1: 4.2.1</td>
</tr>
<tr>
<td></td>
<td>Medication Treatment Planning</td>
<td>PHARM TF-1: 4.2.3</td>
</tr>
<tr>
<td>Pharmaceutical Adviser</td>
<td>Workflow Management</td>
<td>PHARM TF-1: 4.2.4</td>
</tr>
<tr>
<td></td>
<td>Provision of Medication List</td>
<td>PHARM TF-1: 4.2.1</td>
</tr>
<tr>
<td></td>
<td>Medication Treatment Planning</td>
<td>PHARM TF-1: 4.2.3</td>
</tr>
</tbody>
</table>

17 Only applicable, if option “Provision of Medication List” is supported.
4.2.1 Provision of Medication List Option
A Community Pharmacy Manager Actor implementing this option offers the ability to query for a Medication List and return an on-demand created version of the Medication List document. A Medication Treatment Planner-, Prescription Placer-, Pharmaceutical Adviser-, Medication Dispenser Actor implementing this option offers the ability to query for a Medication List and interpret its content according to the PML Profile. See use case described in Volume 1, chapter 4.4.3.

4.2.2 Persistence of Retrieved Documents Option
A Community Pharmacy Manager Actor implementing this option offers the ability to persist the returned on-demand created version of the Medication List document when querying for a Medication List. This option requires the “Provision of Medication List” Option to be implemented. See use case described in Volume 1, chapter 4.4.3 and ITI TF-1: 10.2.7.

4.2.3 Medication Treatment Planning Option
A Community Pharmacy Manager Actor implementing this option offers the ability to query for Medication Treatment Plans and to return the query result. A Prescription Placer-, Pharmaceutical Adviser-, Medication Dispenser Actor implementing this option offers the ability to query for Medication Treatment Plans and interpret their content according to the MTP Profile. See use case described in Volume 1, chapter 4.4.2.

4.2.4 Workflow Management Option
An actor implementing this option offers the ability to manage workflow according to the Workflow Definitions described in Volume 2, chapter 4.

4.3 CMPD Actor Groupings and Profile Interactions

<table>
<thead>
<tr>
<th>Actor</th>
<th>Groups with</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medication Treatment Planner</td>
<td>Content Creator: MTP</td>
<td>The Medication Treatment Planner Actor shall create Medication Treatment Plan documents according to the MTP Content Profile.</td>
</tr>
<tr>
<td></td>
<td>Content Consumer: MTP, PRE, PADV, DIS, PML</td>
<td></td>
</tr>
</tbody>
</table>
4.4 CMPD Process Flow

Current implementations of the community pharmacy process (plan, prescribe & dispense medication) may be categorized in two different alternatives.

The first alternative is the so-called publish & pull. In this model, generally speaking, information is generated by a placer type actor (Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser) and stored by means of a repository type actor. Other actors retrieve data by pulling it from repositories. This approach may apply to health systems where information is accessed on a centralized basis and, therefore, is made available to a collective of potential users (such as prescriptions available for dispense in any community pharmacy).

The alternative approach is the direct push model where information is sent directly to the actor intended to use it (e.g., prescriptions sent directly to the pharmacy named by the patient) and therefore no information is stored on a centralized basis. This model focuses on direct communication instead of availability to (more) potential users.

The current revision of the Integration Profile covers use cases relying on the publish & pull model only.

Workflow scenarios

The CMPD Process Flow can be principally differentiated in two basic workflow scenarios, one including a validation step by a Pharmaceutical Adviser Actor and another excluding it:

- **Scenario 1: Including a validation step by a Pharmaceutical Adviser**
- **Scenario 2: Not including a validation step by a Pharmaceutical Adviser**

A domain using CMPD has to define in which workflow scenario it operates. Workflow scenarios cannot be used compounded.
Any software implementations of the CMPD Profile have to be able to operate in both workflow scenarios.

### 4.4.1 Use Case community pharmacy-active substance, publish & pull (Scenario 1: “Including validation step”)

The purpose of this use case is to illustrate the prescription-dispense process in community pharmacy when the prescriber orders an active-substance (generic) medicine in the publish & pull model. The optional initial planning action is not represented in this scenario in order to limit its complexity. Inclusion of the initial planning step can be seen in the second scenario.

The process of this use case includes the validation step performed by a Pharmaceutical Adviser Actor.

The following diagram shows the workflow of this use case and illustrates the overall context of

<table>
<thead>
<tr>
<th>… (workflow) tasks</th>
<th>Ordering, Validation, Dispensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>… which actor performs the task</td>
<td>Prescription Placer, Pharmaceutical Adviser, Medication Dispenser</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>… the conditions leading to the next task</th>
</tr>
</thead>
<tbody>
<tr>
<td>In some cases depending on the outcome of the pharmaceutical validation documented in a Pharmaceutical Advice document (see PADV Profile)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>… on which task-transition each query of transaction PHARM-1 is used and which business rule it has to follow</th>
</tr>
</thead>
<tbody>
<tr>
<td>FindPrescriptionsForValidation (by the Pharmaceutical Adviser)</td>
</tr>
<tr>
<td>FindPrescriptionsForDispense (by the Medication Dispenser)</td>
</tr>
</tbody>
</table>
This workflow is implicitly specified by the narrative descriptions in both this profile as well as in the Pharmacy Pharmaceutical Advice Profile.

Note that software implementations shall be able to perform it in any case, whether or not actors are grouped with option “Workflow Management” (grouping with that option does not change the actual workflow, it just allows “technical” workflow management).

Please refer to:

- Community Prescription and Dispense (CMPD) Profile
  
  Volume 2, chapter 3.1.4.1.2.1.1.3 FindPrescriptionsForValidation
  
  Volume 2, chapter 3.1.4.1.2.1.1.4 FindPrescriptionsForDispense

  In case of grouping with XDW: Volume 2, chapter 4 Workflow Definitions

- Pharmacy Pharmaceutical Advice (PADV) Profile
  
  Vol. 2, chapter 6.3.4.3.3.6 Status Code
  
  Vol. 2, chapter 6.3.4.3.3.4 Observation Code
4.4.1.1 Story Board

John Doe attends a consultation to his general practitioner, GP, because he is experiencing some breathing difficulty. The practitioner examines John and prescribes the active substance “Fenoterol” in his “Prescription Placer” software. The prescription is electronically sent to the “Prescription Repository”.

Since prescriptions are available to a wide range of pharmacies, John picks the pharmacy closest to his office. The pharmacist asks for John’s health card in order to retrieve the patient’s active prescriptions. Since John also suffers from arthritis he has been prescribed Ibuprofen. The pharmacists checks for interactions and finds nothing outstanding. The information on the pharmaceutical advice is electronically sent to the “Pharmaceutical Advice Repository”.

He consults his inventory and picks Berotec® which is in the range of prices approved by the health system. He gives out this medicine to the patient and records the transaction in the “Medication Dispenser”. The information on the medication dispensed is electronically sent to the “Dispensed Medication Repository”.

4.4.1.2 Sequence Diagram

The following diagram represents the sequence of data exchanged between “system actors” involved in this use case.
Physician prescribes Fenoterol

Patient requests medication delivery

1) Determine Prescription Items to validate
2) Interactions checked; Prescription Item approved (Fenoterol)

Query and retrieve: Prescriptions, Pharm.advises and Dispenses of patient

Pharmaceutical Advice to Fenoterol submitted

Provide and Register Document set [ITI-41]

Dispense to Fenoterol submitted

Provide and Register Document set [ITI-41]

Get Prescriptions for dispensation

Query Pharmacy Documents [PHARM-1]

Retrieve Document Set [ITI-43]

Determine Prescription Items to dispense

Get Prescriptions for validation

Query Pharmacy Documents [PHARM-1]

Retrieve Document Set [ITI-43]

Patient requests medication dispense

530

This diagram illustrates the complete workflow of the prescription of a medication, the successful validation of the Prescription Item and the dispense of the medication (Workflow scenario 1).
4.4.2 Use Case community pharmacy-active substance, plan, publish & pull
(Scenario 2: “Not including validation step”)

The purpose of this use case is to illustrate the planning-prescription-dispense process in
community pharmacy when the prescriber first plans an active substance (generic) medicine and
then orders it in the publish & pull model.

The process of this use case does not include the validation step performed by a Pharmaceutical
Adviser Actor.

The following diagram shows the workflow of this use case and illustrates the overall context of

- … (workflow) tasks
  Planning, Ordering, Dispensing
- … which actor performs the task
  Medication Treatment Planner, Prescription Placer, Medication Dispenser
- … the conditions leading to the next task
- … on which task-transition each query of transaction PHARM-1 is used and which
  business rule it has to follow
  FindMedicationTreatmentPlans (by the Prescription Placer)
  FindPrescriptionsForDispense (by the Medication Dispenser)
This workflow is implicitly specified by the narrative descriptions in this profile. Note that software implementations shall be able to perform it in any case, whether or not actors are grouped with option “Workflow Management” (grouping with that option does not change the actual workflow, it just allows “technical” workflow management).

Please refer to:

- Community Prescription and Dispense (CMPD) Profile
  Volume 2, chapter 3.1.4.1.2.1.1.6 FindMedicationTreatmentPlans
  Volume 2, chapter 3.1.4.1.2.1.1.4 FindPrescriptionsForDispense

In case of grouping with XDW: Volume 2, chapter 4 Workflow Definitions

### 4.4.2.1 Story Board

John Doe attends a consultation to his general practitioner, GP, because he is experiencing some breathing difficulty. The practitioner examines John and decides to add the active substance “Fenoterol” in his “Medication Treatment Plan Planner” software. The planned medication is electronically sent to the “Medication Treatment Plan Repository”. As a prescription is required for getting this medication from the pharmacy, the practitioner also prescribes the active...
substance “Fenoterol” in his “Prescription Placer” software. The prescription is electronically sent to the “Prescription Repository”.

Since prescriptions are available to a wide range of pharmacies, John picks the pharmacy closest to his office. The pharmacist asks for John’s health card in order to retrieve the patient’s active prescriptions.

He consults his inventory and picks Berotec® which is in the range of prices approved by the health system. He gives out this medicine to the patient and records the transaction in the “Medication Dispenser”. The information on the medication dispensed is electronically sent to the “Dispensed Medication Repository”.

4.4.2.2 Sequence Diagram

The following diagram represents the sequence of data exchanged between “system actors” involved in this use case.
Physician plans Fenoterol

Physician prescribes Fenoterol

Determine Planned Medication Item to Prescribe

Patient requests medication dispense

Get Prescriptions for dispense

Query Pharmacy Documents [PHARM-1]

Retrieve Document Set [ITI-43]

Dispense to Fenoterol submitted

Patient requests medication dispense

Get Prescriptions for dispense

Query Pharmacy Documents [PHARM-1]

Retrieve Document Set [ITI-43]

Dispense to Fenoterol submitted

Figure 4.4.2.2-1: Use Case community pharmacy-active substance, publish & pull - Process Flow (Scenario 2: “Not including validation step”)

This diagram illustrates the complete workflow of the planning, prescription and dispense of the medication without validation (Workflow Scenario 2).
4.4.3 Use Case: Physician requests Medication List

The purpose of this use case is to illustrate the process of requesting the Medication List by a physician. This requires the support of the “Provision of Medication List” Option at the Community Pharmacy Manager.

4.4.3.1 Story Board

John Doe attends a consultation to his general practitioner, GP, because he is experiencing some breathing difficulty. The practitioner examines John and wants to prescribe the active substance “Fenoterol” in his “Prescription Placer” software.

To ensure that there are no conflicts between the new medication and the patient’s current medication status, the physician requests the Medication List.

The Prescription Placer Actor uses transaction “Query Pharmacy Document” (PHARM-1) with query “FindMedicationList” to query the Medication List at the Community Pharmacy Manager Actor.

The Community Pharmacy Manager queries the registry for the on-demand document entry of the Medication List to this patient. In case this query does not return a valid result the Community Pharmacy Manager uses transaction “Register On-Demand Document” (ITI-61) to register the Medication List On-Demand Document. Either the found or just created Document Entry will be returned to the calling Prescription Placer Actor.

The Prescription Placer Actor now uses transaction “Retrieve Document Set” (ITI-43) to retrieve the actual Medication List Document from the Community Pharmacy Manager. The Community Pharmacy Manager Actor uses ITI XDS transactions to query and retrieve Medication Treatment Plan-, Prescription-, Pharmaceutical Advice- and Dispense Documents from the repositories in order to assemble the Medication List Document. Once the document is assembled it returns the document to the calling Prescription Placer Actor. If the “Persistence of Retrieved Documents” Option is used the returned document is also provided and registered in the registry/repository backend.
4.4.3.2 Sequence Diagram
The following diagram represents the sequence of data exchanged between “system actors” involved in this use case.

Figure 4.4.3-1: Use Case Physician requests Medication List - Process Flow
This diagram illustrates the complete workflow of the process of requesting the Medication List.
4.4.4 Use Case: Physician changes/cancels or suspends an unfilled prescription

The purpose of this use case is to illustrate the process of changing/canceling or suspending an unfilled prescription by a physician.

4.4.4.1 Story Board

After getting Fenoterol prescribed by a physician, the patient, not having the medication dispensed by the pharmacy yet, re-visits the physician on the next day because the illness had improved. The patient has not yet received the prescribed medication at a pharmacy.

The physician performs another physical examination to confirm the improved health status and decides to amend the original prescription of Fenoterol by either changing it (e.g., to keep the medication, but with a lower dosage), canceling it (because it’s not needed anymore) or suspend it (to observe the further course of the illness with the intention to decide later if the medication shall be given or canceled).

The physician issues a Pharmaceutical Advice document to record the command and instructs the patient.

4.4.4.2 Sequence Diagram

The following diagram represents the sequence of data exchanged between “system actors” involved in this use case.
4.4.5 Use Case: Physician changes/stops or suspends the treatment with an already dispensed medication

The purpose of this use case is to illustrate the process of changing/stopping or suspending the treatment with an already dispensed medication by a physician.

4.4.5.1 Story Board

After getting a prescription for Fenoterol by a physician, the patient has the medication dispensed by the pharmacy. The patient takes the medication for seven days. He then re-visits the physician because the illness had improved.

The physician performs another physical examination to confirm the improved health status and decides to amend the treatment with Fenoterol by either changing it (e.g., with a lower dosage), stopping it (because it’s not needed anymore) or suspending it (e.g., to observe the further course of the illness with the intention to decide later if the medication shall be continued or stopped).18

18 Setting a medication treatment to suspend might also be used at admission of the patient into a hospital, because the hospital takes full control over the medication of the patient during the hospital stay. The original medication might be resumed at discharge of the patient.
The physician issues a Pharmaceutical Advice document to record the command and instructs the patient.

### 4.4.5.2 Sequence Diagram

The following diagram represents the sequence of data exchanged between “system actors” involved in this use case.

![Sequence Diagram](image)

**Figure 4.4.5-1: Use Case: Physician changes/stops or suspends the treatment with an already dispensed medication**

### 4.5 CMPD Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the XDS Security Considerations Section (see ITI TF-1: 10.7).

### 4.6 CMPD Implementation Scenarios

The following chapter describes several implementation scenarios for the Community Prescription and Dispense Integration Profile.

The planning, prescription and dispense process of real-world projects involves several parties acting in the different abstract roles (Medication Treatment Planner, Prescription Placer,
Pharmaceutical Adviser, Medication Dispenser). The Medication Treatment Planner and Prescription Placer roles are usually taken by physicians; the Pharmaceutical Adviser and Medication Dispenser role is usually taken by pharmacists, which both are usually organized in different organizations.

This results in a wide variety of implementation requirements together with the need of not only organizational but also technical separation of systems. Physicians may want to store plans and prescriptions in another repository than pharmacists the dispenses. In a strict separation even the use of separate IHE affinity domains is required to arrange a throughout distinct scenario. CMPD was designed to be used in either *single-domain* or *multi-domain* scenarios to fit to these requirement.

Any political intended separation has to be technically bridged at one point otherwise a common planning, prescription and dispense process cannot be established. To minimize the possible points of contact between the domains the Community Pharmacy Manager was introduced.

Explanation to the diagrams used in the following implementation scenario chapters:

- Dotted lines mean separation of concerns
- Different background colors mean different XDS affinity domains

### 4.6.1 Usage of CMPD in a “single-domain” scenario

The descriptions of CMPD in the previous chapters are aligned to the usage of the profile in a scenario where all actors are hosted in a *single XDS Affinity domain*.

Operating within a single XDS Affinity domain is the most simple implementation scenario and has several benefits, like e.g., that just one registry holds any document metadata, which eases query, retrieving and publishing of documents, etc.

On the other hand a simple scenario like this may not be applicable to scenarios in reality, where organizational, strategical or political reasons require more separation between the participating parties (physicians, pharmacists).

The following diagram shows a simple example of a single-domain implementation scenario to demonstrate the capabilities of CMPD.
Description of the example scenario

The group of Medication Treatment Planners, Prescription Placers, Pharmaceutical Advisers and Medication Dispensers are altogether located in one XDS affinity domain. Each group stores its documents in its own dedicated repository, but all use the same document registry of the affinity domain.¹⁹

¹⁹ In an even more simplified scenario the different document repositories could be merged into one single document repository, but this would not change the principles of the example.
4.6.1.1 Demonstration of use case 1 in example scenario (simple)

Step 1: Medication Treatment Planner creates a plan

The Medication Treatment Plan is submitted to the appropriate Medication Treatment Plan repository.
Step 2: Prescription Placer queries the plan

The Prescription Placer queries the plan by using transaction PHARM-1, query “FindMedicationList” in order to retrieve active Medication Treatment Plan Items.

The CPM queries the common XDS domain registry for medication treatment plan and pharmaceutical advice documents. Then it retrieves all these documents from the appropriate document repositories.

After retrieving it does linking of the documents by their document IDs and determines the status of each medication treatment plan. It applies appropriate filtering according to the semantic question “Ready for prescription” (i.e., “Active”) and returns just “relevant” document UUIDs to the Prescription Placer Actor, which proceeds with step 3.

Step 3: Prescription Placer retrieves the documents of the query result

The Prescription Placer Actor asks the CPM to retrieve all documents identified by the returned document UUIDs from the according document repositories.

The Prescription Placer Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the system or the human operator performs the selection of medication treatment plans to prescribe and proceeds with step 4.
Step 4: Prescription Placer creates a prescription
The Prescription document is submitted to the appropriate Prescription Repository.
Step 5: Pharmaceutical Adviser queries the prescription

The Pharmaceutical Adviser queries the prescription by using transaction PHARM-1, query “FindPrescriptionsForValidation”.

The CPM queries the common XDS domain registry for prescription, pharmaceutical advice and dispense documents. Then it retrieves all these documents from the appropriate document repositories.

After retrieving it does linking of the documents by their document IDs and determines the status of each prescription. It applies appropriate filtering according to the semantic question “for Validation” and returns just “relevant” document UUIDs to the Pharmaceutical Adviser Actor, which proceeds with step 6.
Step 6: Pharmaceutical Adviser retrieves the documents of the query result

The Pharmaceutical Adviser Actor asks the CPM to retrieve all documents identified by the returned document UUIDs from the according document repositories.

The Pharmaceutical Adviser Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the system or the human operator performs validation and proceeds with step 7.
Step 7: Pharmaceutical Adviser submits a pharmaceutical advice

After the validation step the outcome of the validation is documented in a Pharmaceutical Advice document. This document is submitted to the appropriate Pharmaceutical Advice Repository.
Step 8: Medication Dispenser queries the prescription

The Medication Dispenser queries the prescription by using transaction PHARM-1, query “FindPrescriptionsForDispense”.

Analog to step 5, the CPM queries the XDS domain registry for prescription, pharmaceutical advice and dispense documents. Then it retrieves all these documents from the appropriate document repositories.

After retrieving it performs linking of the documents by their document IDs and determines the status. It applies appropriate filtering according to the semantic question “for Dispense”, and returns just “relevant” document UUIDs to the Medication Dispenser Actor, which proceeds with step 9.
Step 9: Medication Dispenser retrieves the documents of the query result

The Medication Dispenser Actor asks the CPM to retrieve all documents identified by the returned document UUIDs from the according document repositories.

The Medication Dispenser Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the human operator performs the dispense and proceeds with step 10.
Step 10: Medication Dispenser submits a dispense

After the dispense has taken place it is documented in a Dispense document. This document is submitted to the appropriate Dispensed Medication Repository.
4.6.2 Usage of CMPD in a “multi-domain” scenario

The descriptions of CMPD in the previous chapters are aligned to the usage of the profile in a scenario where all actors are hosted in a single XDS Affinity domain. Nevertheless, the profile can also be used in multi XDS Affinity domain scenarios.

Operating within a scenario consisting of multiple XDS Affinity domains is a complex but rather realistic implementation scenario. Its main benefit is that a minimum of technical contact is required between the participating parties of such a system (physicians, pharmacists) for achieving technical interoperability. Such utmost separation might be an organizational, strategical or political requirement.

Such an implementation scenario requires the usage of the CPM’s “relaying” functionality shown in the following more detailed Actor Diagram:

![Actor Diagram](image-url)

**Figure 4.6.2-1: More detailed Actor Diagram in a multi-domain scenario**
The following diagram shows an example of a possible multi-domain implementation scenario to demonstrate the capabilities of CMPD.

Note that as the “Plan” level (Medication Treatment Planner) does not really affect the workflow of this scenario and is not mandatory, it is not included here. However a structure similar to the previous scenario could also have been described.

**Description of the example scenario**

The group of Prescription Placers divides into 2 separate domains, the first (PRE1) showing a federated architecture with multiple repositories, the second (PRE2) with all clients connected to one.

The group of Pharmaceutical Advisers is organized in an own affinity domain (PADV), all storing in one repository.

The group of Medication Dispensers are all organized in a common affinity domain (DIS), but everyone stores its dispenses in their own application (also acting as repository).

*All these different domains accept the XDS Affinity domain of the Community Pharmacy Manager (CPM) as the point of intersection to which they all maintain trusted relationships to. Bi-lateral trusts are not required.*
4.6.2.1 Demonstration of use case 1 in example scenario (complex)

Step 1: Prescription Placer creates a prescription
The Prescription document is submitted to the appropriate Prescription Repository.
Step 2: Pharmaceutical Adviser queries the prescription

The Pharmaceutical Adviser queries the prescription by using transaction PHARM-1, query “FindPrescriptionsForValidation”.

In this complex scenario the CPM has to use XCA mechanisms to query all other domains for prescription, pharmaceutical advice and dispense documents. Then it retrieves all these documents from the appropriate document repositories.

After retrieving it does linking of the documents by their document IDs and determines the status of each prescription. It applies appropriate filtering according to the semantic question “for Validation” and returns just “relevant” document UUIDs to the Pharmaceutical Adviser Actor, which proceeds with step 3.
870 Step 3: Pharmaceutical Adviser retrieves the documents of the query result

The Pharmaceutical Adviser Actor asks the CPM to retrieve all documents identified by the returned document UUIDs.

The CPM acts as a relaying entity and accesses all requested repositories for retrieving the documents by XCA. Then it returns them to the calling client. Note that the Pharmaceutical Adviser Actor has no need to have access to the other domains (which could be organizational prohibited).

The Pharmaceutical Adviser Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the system or the human operator performs validation and proceeds with step 4.
Step 4: Pharmaceutical Adviser submits a pharmaceutical advice

After the validation step the outcome of the validation is documented in a Pharmaceutical Advice document. This document is submitted to the appropriate Pharmaceutical Advice Repository.
Step 5: Medication Dispenser queries the prescription

The Medication Dispenser queries the prescription by using transaction PHARM-1, query “FindPrescriptionsForDispense”.

Analog to step 2, the CPM uses XCA mechanisms to query all other domains for prescription, pharmaceutical advice and dispense documents. Then it retrieves all these documents from the appropriate document repositories.

After retrieving it performs linking of the documents by their document IDs and determines the status. It applies appropriate filtering according to the semantic question “for Dispense”, and returns just “relevant” document UUIDs to the Medication Dispenser Actor, which proceeds with step 6.
Step 6: Medication Dispenser retrieves the documents of the query result

The Medication Dispenser Actor asks the CPM to retrieve all documents identified by the returned document UUIDs.

The CPM acts as a relaying entity and accesses all requested repositories for retrieving the documents by XCA. Then it returns them to the calling client. Note that the Medication Dispenser Actor has no need to have access to the other domains (which could be organizational prohibited).

The Medication Dispenser Actor (as a machine) parses and relinks the returned documents by their document IDs. Then the human operator performs the dispense and proceeds with step 7.
Step 7: Medication Dispenser submits a dispense

After the dispense has taken place it is documented in a Dispense document. This document is submitted to the appropriate Dispensed Medication Repository.
Appendix A Actor Summary Definitions

**Community Pharmacy Manager** - Actor providing the business logic for status management and other purposes. As a second role it acts as a “relaying role” where certain standard XDS communication is routed through for providing the possibility of applying project-specific business logic on it.

**Medication Treatment Planner** - Actor for planning a new medication (introducing a new medication into the patient’s treatment plan). It provides Medication Treatment Plan documents each containing one Medication Treatment Plan Item representing the planned medication.

**Prescription Placer** - Actor for placing prescriptions (initial or modified in case of a substitution of invalidation, for example). It provides Prescription documents containing one or more Prescription Items representing the prescribed medication.

**Pharmaceutical Adviser** - Actor responsible for the validation of prescriptions from a pharmacist’s perspective. It sends provides the Pharmaceutical Advice document as the result of the validation. Pharmaceutical Advisers (e.g., automated ICA check modules) may also provide “draft” advices which don’t affect the status of a prescription but serve as a foundation for the advice performed by another Pharmaceutical Adviser.

**Medication Dispenser** - Actor responsible for the process of dispensing medication to the patient, fulfilling the prescription. It receives prescriptions already validated and provides a dispense document as result of the act of delivering the medication to the patient.

**Registry/Repository actors** - Formally the Community Pharmacy process defines three different “repositories” for Prescriptions, Pharmaceutical Advices and Dispenses. They shall be seen as abstract repository-roles for persisting the appropriate document types the documents. This profile makes use of the XDS Profile for defining abstract XDS registry and repository actors for modeling the abstract repository-roles for real implementations.

Appendix B Transaction Summary Definitions

**Query Pharmacy Documents** - This transaction defines how a querying actor has to query the Community Pharmacy Manager for medication treatment plans (MTP) and prescriptions (PRE) and their related documents. Related documents are Pharmaceutical Advice (PADV) and Dispense (DIS) documents. It defines specialized queries to allow the finding of plans or prescriptions and their related documents for specific purposes (e.g., “for validation” or “for dispense”).

**Registry Stored Query** - See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2a:3.18)

**Provide and Register Document Set-b** - See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2a:3.41)
Retrieve Document Set - See the XDS Integration Profile of the ITI Technical Framework for a detailed description of this transaction (ITI-TF2a:3.43)
3.1 Query Pharmacy Documents [PHARM-1]

This transaction defines how a querying actor has to query the Community Pharmacy Manager for medication treatment plans (MTP), prescriptions (PRE) and their related documents. Related documents are Pharmaceutical Advice (PADV) and Dispense (DIS) documents.

Specialized queries allow the finding of medication treatment plans, prescriptions and their related documents for specific purposes (e.g., for prescription or for validation).

Querying actors may be:
- Medication Treatment Planner
- Prescription Placer
- Pharmaceutical Adviser
- Medication Dispenser

This transaction is very similar to the concept of the Registry Stored Query (ITI-18) transaction in the XDS Integration Profile of the ITI Technical Framework, except that the query itself targets not a single registry (like described in the XDS Integration Profile) but shall be able to sub-query one to many registry/repository systems (by using XCA in case of multi-domain scenarios) to get the requested query result.

The querying actor faces the same interface as if querying a XDS Document registry actor, although the query result may contain references to documents of many different domains.

3.1.1 Scope

The Query Pharmacy Documents transaction supports the following specialized queries:

- **FindPrescriptions**
  Find prescriptions and their related documents

- **FindDispenses**
  Find dispense documents and their related documents

- **FindPrescriptionsForValidation**
  Find prescriptions and their related documents containing Prescription Items ready to be validated
• **FindPrescriptionsForDispense**
  Find prescriptions and their related documents containing Prescription Items ready to be dispensed

1005
• **FindMedicationList (if „Provision of Medication List“ Option is supported)**
  Find the medication list to the patient

• **FindMedicationTreatmentPlans (if „Medication Treatment Planning“ Option is supported)**
  Find planned medications and their related documents

1010
All queries return:
• Metadata for one or more registry objects, or
• Object references for one or more registry objects (registry UUIDs).

### 3.1.2 Use Case Roles

**Actors:** Querying actor

**Role:** Requests a query by identifier (UUID), and passes parameters to the query. A parameter controlling the format of the returned data is passed; it selects either object references or full objects.

**Actor:** Community Pharmacy Manager

**Role:** Services the query using its stored definitions of the queries defined for CMPD.

### 3.1.3 Referenced Standard

ITI-18: Registry Stored Query and all its related standards.
3.1.4 Interaction Diagram

3.1.4.1 Query Pharmacy Documents

This is a query request to the Community Pharmacy Manager from a Querying actor. The query request contains:

- A reference to a pre-defined query stored on the Document Registry Actor.
- Parameters to the query. The query parameters are matched up with the query variables defined in the query definition on the Document Registry Actor.

3.1.4.1.1 Trigger Events

This message is initiated when the Querying actor wants to query/retrieve document metadata.

This may be the case, if:

1. A Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser wants to find medication treatment plans, prescriptions or dispenses (and their related pharmaceutical advices).

2. A Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser wants to find dispense documents (and their related pharmaceutical advices, plans and prescriptions).

3. A Pharmaceutical Adviser Actor wants to find active prescriptions (and their related pharmaceutical advices and dispenses) ready to validate.

4. A Medication Dispenser wants to find active plans or prescriptions (and their related pharmaceutical advices and dispenses) which are already validated or ready for dispense.

5. A Medication Treatment Planner, Prescription Placer, Pharmaceutical Adviser or Medication Dispenser wants to find the Medication List to the patient.
3.1.4.1.2 Message Semantics

The message semantics of this message are based on the definitions of the [ITI-18] transaction, but incorporate some important changes defined in the chapters below.

References to: ITI TF-2a: [ITI-18]

3.1.4.1.2.1 Required Queries

The Registry Stored Query (ITI-18) transaction defines several kinds of Stored Queries (FindDocuments, FindSubmissionSets, etc.). The PHARM-1 transaction is alike to this concept but provides a different set of Stored Queries.

The provided Stored Queries are:

- **FindPrescriptions**
  Find prescriptions and their related documents

- **FindDispenses**
  Find dispense documents and their related documents

- **FindPrescriptionsForValidation**
  Find prescriptions and their related documents containing Prescription Items ready to be validated

- **FindPrescriptionsForDispense**
  Find prescriptions and their related documents containing Prescription Items ready to be dispensed

- **FindMedicationList (if “Provision of Medication List“ Option is supported)**
  Find the medication list to the patient

- **FindMedicationTreatmentPlans (if “Medication Treatment Planning” Option is supported)**
  Find planned medications and their related documents

3.1.4.1.2.1.1 Parameters for Required Queries

This chapter defines the parameters and business rules for the Required Queries.

---

Note: Some business rules include matching rules to FormatCodes (e.g., “FormatCode matches urn:ihe:pharm:pre:2010”). Projects leveraging IHE Pharmacy profiles may specify additional constraints on the profiles resulting in project-specific templates. This may include the definition of project-specific
FormatCodes for the documents. The business rules may be adjusted to match to such project-specific FormatCodes.

Example: If a project uses own FormatCodes, e.g., “urn:project:prescription:2014” instead of “urn:ihe:pharm:pre:2010” for prescriptions, it may use those in replacement to the ones defined in the profile.

### 3.1.4.1.2.1.1.1 FindPrescriptions

Find prescriptions and their related documents (XDSDocumentEntry objects) containing Prescription Items for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

**Returns:** XDSDocumentEntry objects according to the following business rules.

**Business rule 1:** Returns *Prescription* documents matching the query parameters:
- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters
- FormatCode matches **urn:ihe:pharm:pre:2010**

**Business rule 2:** Returns related *Dispense* documents to the Prescriptions found by business rule 1
- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:dis:2010**
- Dispense document contains a Dispense Entry Item related to a Prescription Item of the found Prescription documents.

**Business rule 3:** Returns related *Medication Treatment Plan* documents to the Prescriptions found if Medication Treatment Plan Planner is used
- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- Medication Treatment Plan document contains a Medication Treatment Plan Entry Item related to a Prescription Item of the found Prescription documents.

---

20 See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)
Business rule 4: Returns related *Pharmaceutical Advice* documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches `urn:ihe:pharm:padv:2010`
- Pharmaceutical Advice document contains
  - (1) a Pharmaceutical Advice Entry Item related to a Prescription Item of the found Medication Treatment Plan documents
  - (2) a Pharmaceutical Advice Entry Item related to a Medication Treatment Plan Item of the found Medication Treatment Plan documents

**Explanation**

Returning Prescription documents according to business rule 1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Medication Treatment Plans, Pharmaceutical Advice and Dispense documents to the Prescriptions (Business rule 2 to 4), is dependent on the primary result (found Prescriptions) only and contains just directly related documents.

**Query parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Attribute</th>
<th>Opt</th>
<th>Mult</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XDSDocumentEntryPatientId</td>
<td>XDSDocumentEntry. patientId</td>
<td>R</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryEntryUUID</td>
<td>XDSDocumentEntry. entryUUID</td>
<td>O¹</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryUniqueId</td>
<td>XDSDocumentEntry. uniqueId</td>
<td>O¹</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryPracticeSettingCode³</td>
<td>XDSDocumentEntry. practiceSettingCode</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryCreationTimeFrom</td>
<td>Lower value of XDSDocumentEntry. creationTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryCreationTimeTo</td>
<td>Upper value of XDSDocumentEntry. creationTime</td>
<td>O</td>
<td>--</td>
</tr>
</tbody>
</table>

21 See the Pharmacy Prescription Content Profile (MTP) for details how the relation between Medication Treatment Plan Entries and Prescription Items is defined, chapter: “Prescription Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.2)

22 See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Medication Treatment Plan-/Prescription Items is defined, chapter: “Pharmaceutical Advice Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.3)
### Parameter Name | Attribute | Opt | Mult
--- | --- | --- | ---
$XSDocumentEntryServiceStartTimeFrom | Lower value of XSDocumentEntry. serviceStartTime | O | --
$XSDocumentEntryServiceStartTimeTo | Upper value of XSDocumentEntry. serviceStartTime | O | --
$XSDocumentEntryServiceStopTimeFrom | Lower value of XSDocumentEntry. serviceStopTime | O | --
$XSDocumentEntryServiceStopTimeTo | Upper value of XSDocumentEntry. serviceStopTime | O | --
$XSDocumentEntryHealthcareFacilityTypeCode | XSDocumentEntry. healthcareFacilityTypeCode | O | M
$XSDocumentEntryEventCodeList | XSDocumentEntry. eventCodeList | O | M
$XSDocumentEntryConfidentialityCode | XSDocumentEntry. confidentialityCode | O | M
$XSDocumentEntryAuthorPerson | XSDocumentEntry. Author | O | M
$XSDocumentEntryStatus | XSDocumentEntry. Status | R | M

1Either $XSDocumentEntryEntryUUID or $XSDocumentEntryUniqueld shall be specified. This transaction shall return an error if both parameters are specified.

3Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

4The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute)

### Examples for the “FindPrescriptions” query

Assume the following situation of persisted documents in the Prescription-/Pharmaceutical Advice- and Dispense repositories:

<table>
<thead>
<tr>
<th>Treatment Plans</th>
<th>Prescriptions</th>
<th>Pharmaceutical Advice</th>
<th>Dispenses</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTP 1</td>
<td>PRE 1</td>
<td>PRE Item 1-1</td>
<td>PADV 1</td>
<td>DIS 1</td>
</tr>
<tr>
<td>MTP 2</td>
<td>PRE Item 1-2</td>
<td>PADV 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTP 3</td>
<td>PRE 2</td>
<td>PRE Item 2-1</td>
<td>PADV 3</td>
<td>DIS 2</td>
</tr>
</tbody>
</table>
Note that Medication Treatment Plan Items (documents) are optional and may not be present. They are included in the examples for describing the situation where a Medication Treatment Plan Planner is being used.

**Example 1: Standard query**

Used Query Parameters:
- Patient ID
- Document Status

This is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prescriptions</td>
<td>Related documents</td>
</tr>
<tr>
<td>PRE 1</td>
<td>MTP 1, MTP 2, PADV 1, PADV 2, DIS 1</td>
</tr>
<tr>
<td>PRE 2</td>
<td>MTP 3, MTP 4, MTP 5, PADV 3, PADV 4, DIS 2</td>
</tr>
<tr>
<td>PRE 3</td>
<td>MTP 6</td>
</tr>
</tbody>
</table>

**Example 2: Search for a specific prescription**

Query Parameters set:
- Patient ID
- Document Status
- Document uniqueld of the specific prescription (e.g., because patient showed a paper prescription with the uniqueld printed on it)

In case the uniqueld of PRE 2 is given as query parameter, this is what should be returned by the query:
3.1.4.1.2.1.1.2 FindDispenses

Find dispense documents and their related documents (XDSDocumentEntry objects) containing Dispense Items for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

Returns: XDSDocumentEntry objects according to the following business rules.

Business rule 1: Returns Dispense documents matching the query parameters:
- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters

Business rule 2: Returns related Prescription documents to the Dispense documents found
- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches urn:ihe:pharm:pre:2010
- Prescription document contains a Prescription Item on behalf of which the found Dispense Item has been dispensed.23

Business rule 3: Returns related Medication Treatment Plan documents to the Dispense documents found if Medication Treatment Plan Planner is used
- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- Medication Treatment Plan document contains:
  (1) a Medication Treatment Plan Entry Item related to a Dispense24 Item of the found Dispense documents

23 See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)
(2) a Medication Treatment Plan Entry Item related to a Prescription Item on behalf the Dispense Item has been dispensed of the found Prescription documents.

Business rule 4: Returns related Pharmaceutical Advice documents to the Dispense documents found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches urn:ihe:pharm:padv:2010
- Pharmaceutical Advice document contains
  (1) a Pharmaceutical Advice Entry Item related to a Dispense Item of the found Dispense documents
  (2) a Pharmaceutical Advice Entry Item related to a Prescription Item on behalf the Dispense Item has been dispensed of the found Prescription documents
  (3) a Pharmaceutical Advice Entry Item related to a Medication Treatment Plan Item of the found Medication Treatment Plan documents

Explanation

Returning Dispense documents according to business rule 1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Pharmaceutical Advice and Prescription documents to the Dispense documents (Business rule 2 to 4), is dependent on the primary result (found Dispense) only and contains just directly related documents.

Query parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Attribute</th>
<th>Opt</th>
<th>Mult</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XDSDocumentEntryPatientId</td>
<td>XDSDocumentEntry. patientId</td>
<td>R</td>
<td>--</td>
</tr>
</tbody>
</table>

24 See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Medication Treatment Plan Entries and Dispense Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)

25 See the Pharmacy Prescription Content Profile (PRE) for details how the relation between Medication Treatment Plan Entries and Prescription Items is defined, chapter: “Prescription Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.2)

26 See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Medication Treatment Plan-/Prescription-/Dispense Items is defined, chapter: “Pharmaceutical Advice Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.3)
### Parameter Name | Attribute | Opt | Mult
---|---|---|---
$XDSDocumentEntryEntryUUID | XDSDocumentEntry. entryUUID | O\(^1\) | M
$XDSDocumentEntryUniqueId | XDSDocumentEntry. uniqueId | O\(^1\) | M
$XDSDocumentEntryPracticeSettingCode\(^3\) | XDSDocumentEntry. practiceSettingCode | O | M
$XDSDocumentEntryCreationTimeFrom | Lower value of XDSDocumentEntry. creationTime | O | --
$XDSDocumentEntryCreationTimeTo | Upper value of XDSDocumentEntry. creationTime | O | --
$XDSDocumentEntryServiceStartTimeFrom | Lower value of XDSDocumentEntry. serviceStartTime | O | --
$XDSDocumentEntryServiceStartTimeTo | Upper value of XDSDocumentEntry. serviceStartTime | O | --
$XDSDocumentEntryServiceStopTimeFrom | Lower value of XDSDocumentEntry. serviceStopTime | O | --
$XDSDocumentEntryServiceStopTimeTo | Upper value of XDSDocumentEntry. serviceStopTime | O | --
$XDSDocumentEntryHealthcareFacilityTypeCode\(^3\) | XDSDocumentEntry. healthcareFacilityTypeCode | O | M
$XDSDocumentEntryEventCodeList\(^1\) | XDSDocumentEntry. eventCodeList\(^1\) | O | M
$XDSDocumentEntryConfidentialityCode\(^3\) | XDSDocumentEntry. confidentialityCode\(^3\) | O | M
$XDSDocumentEntryAuthorPerson\(^4\) | XDSDocumentEntry. Author | O | M
$XDSDocumentEntryStatus | XDSDocumentEntry. Status | R | M

\(^{1}\) Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

\(^{3}\) Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

\(^{4}\) The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute).

#### Examples for the “FindDispense” query
Assume the following situation of persisted documents in the Prescription-/Pharmaceutical Advice- and Dispense repositories:
### Treatment Plans

<table>
<thead>
<tr>
<th>MTP 1</th>
<th>PRE 1</th>
<th>PRE Item 1-1</th>
<th>PADV 1</th>
<th>DIS 1</th>
<th>This item is already validated and dispensed</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTP 2</td>
<td></td>
<td>PRE Item 1-2</td>
<td>PADV 2</td>
<td></td>
<td>This item is already validated and ready for dispense</td>
</tr>
<tr>
<td>MTP 3</td>
<td>PRE 2</td>
<td>PRE Item 2-1</td>
<td>PADV 3</td>
<td>DIS 2</td>
<td>This item is already validated and dispensed</td>
</tr>
<tr>
<td>MTP 4</td>
<td></td>
<td>PRE Item 2-2</td>
<td>PADV 4</td>
<td></td>
<td>This item is already validated and ready for dispense</td>
</tr>
<tr>
<td>MTP 5</td>
<td></td>
<td>PRE Item 2-3</td>
<td></td>
<td></td>
<td>This item is not validated yet</td>
</tr>
<tr>
<td>MTP 6</td>
<td>PRE 3</td>
<td>PRE Item 3-1</td>
<td></td>
<td></td>
<td>This item is not validated yet</td>
</tr>
</tbody>
</table>

Note that Medication Treatment Plan Items (documents) are optional and may not be present. They are included in the examples for describing the situation where a Medication Treatment Plan Planner is being used.

#### Example 1: Standard query

Used Query Parameters:
- Patient ID
- Document Status

This is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prescriptions</strong></td>
<td><strong>Related documents</strong></td>
</tr>
<tr>
<td>DIS 1</td>
<td>MTP 1, MTP 2, PADV 1, PADV 2, PRE 1</td>
</tr>
<tr>
<td>DIS 2</td>
<td>MTP 3, MTP 4, MTP 5, PADV 3, PADV 4, PRE 2</td>
</tr>
</tbody>
</table>

#### Example 2: Search for a specific prescription

Query Parameters set:
- Patient ID
- Document Status
- Document uniqueId of the specific dispense
In case the uniqueId of DIS 2 is given as query parameter, this is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prescriptions</strong></td>
<td>Related documents</td>
</tr>
<tr>
<td>DIS 2</td>
<td>MTP 3, MTP 4, MTP 5, PADV 3, PADV 4, PRE 2</td>
</tr>
</tbody>
</table>

### 3.1.4.1.2.1.1.3 FindPrescriptionsForValidation

Find prescriptions and their related documents (XDSDocumentEntry objects) containing Prescription Items *ready to be validated* for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

**Returns:** XDSDocumentEntry objects according to the following business rules.

The business rules are basically depending on the workflow scenario used (see Vol. 1, chapter 4.4 CMPD Process Flow).

#### Scenario 1 “Including validation step”:

**Business rule 1.1:** Returns *Prescription* documents matching the query parameters:

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters
- FormatCode matches `urn:ihe:pharm:pre:2010`
- Prescription document contains at least one Prescription Item ready to validate
  
  A Prescription Item is ready to validate if there exists no Pharmaceutical Advice Item related to it which has statusCode set to “completed”\(^{27}\).

**Business rule 1.2:** Returns related *Dispense* documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- Dispense document contains a Dispense Entry Item related to a Prescription Item of the found Prescription documents.\(^{28}\)

---

\(^{27}\) See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the statusCode element (chapter “Status Code”)

\(^{28}\)
Business rule 1.3: Returns related **Medication Treatment Plan** documents to the Prescriptions found if Medication Treatment Plan Planner is used

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- Medication Treatment Plan document contains a Medication Treatment Plan Entry Item related to a Prescription Item of the found Prescription documents.\(^{29}\)

Business rule 1.4: Returns related **Pharmaceutical Advice** documents to the Prescriptions documents found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches **urn:ihe:pharm:padv:2010**
- Pharmaceutical Advice\(^{30}\) document contains
  1. Pharmaceutical Advice document contains a Pharmaceutical Advice Entry Item related to a Prescription Item of the found Prescription documents
  2. a Pharmaceutical Advice Entry Item related to a Medication Treatment Plan Item of the found Medication Treatment Plan documents

**Explanation**

Returning Prescription documents according to business rule 1.1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Pharmaceutical Advice and Dispense documents to the Prescriptions (Business rule 1.2 to 1.4), is dependent on the primary result (found Prescriptions) only and contains just directly related documents.

**Scenario 2 “Not including validation step”:**

\(^{28}\) See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)

\(^{29}\) See the Pharmacy Prescription Content Profile (MTP) for details how the relation between Medication Treatment Plan Entries and Prescription Items is defined, chapter: “Prescription Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.2)

\(^{30}\) See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Medication Treatment Plan-/Prescription-/Dispense Items is defined, chapter: “Pharmaceutical Advice Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.3)
Since this scenario does not include a validation step, this query will not be used.

**Query parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Attribute</th>
<th>Opt</th>
<th>Mult</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XDSDocumentEntryPatientId</td>
<td>XDSDocumentEntry. patientId</td>
<td>R</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryEntryUUID</td>
<td>XDSDocumentEntry. entryUUID</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryUniqueId</td>
<td>XDSDocumentEntry. uniqueId</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryPracticeSettingCode3</td>
<td>XDSDocumentEntry. practiceSettingCode</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryCreationTimeFrom</td>
<td>Lower value of XDSDocumentEntry. creationTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryCreationTimeTo</td>
<td>Upper value of XDSDocumentEntry. creationTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStartTimeFrom</td>
<td>Lower value of XDSDocumentEntry. serviceStartTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStartTimeTo</td>
<td>Upper value of XDSDocumentEntry. serviceStartTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStopTimeFrom</td>
<td>Lower value of XDSDocumentEntry. serviceStopTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStopTimeTo</td>
<td>Upper value of XDSDocumentEntry. serviceStopTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryHealthcareFacilityTypeCode3</td>
<td>XDSDocumentEntry. healthcareFacilityTypeCode</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryEventTypeCodeList3</td>
<td>XDSDocumentEntry. eventCodeList3</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryConfidentialityCode3</td>
<td>XDSDocumentEntry. confidentialityCode3</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryAuthorPerson4</td>
<td>XDSDocumentEntry. Author</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryStatus</td>
<td>XDSDocumentEntry. Status</td>
<td>R</td>
<td>M</td>
</tr>
</tbody>
</table>

1Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

3Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

4The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute)
Examples for the “FindPrescriptionsForValidation” query

Assume the following situation of persisted documents in the Prescription-/Pharmaceutical Advice- and Dispense repositories:

<table>
<thead>
<tr>
<th>Treatment Plans</th>
<th>Prescriptions</th>
<th>Pharmaceutical Advice</th>
<th>Dispenses</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTP 1</td>
<td>PRE 1</td>
<td>PADV 1</td>
<td>DIS 1</td>
<td>This item is already validated and dispensed</td>
</tr>
<tr>
<td>MTP 2</td>
<td>PRE Item 1-2</td>
<td>PADV 2</td>
<td></td>
<td>This item is already validated and ready for dispense</td>
</tr>
<tr>
<td>MTP 3</td>
<td>PRE 2</td>
<td>PADV 3</td>
<td>DIS 2</td>
<td>This item is already validated and dispensed</td>
</tr>
<tr>
<td>MTP 4</td>
<td>PRE Item 2-2</td>
<td>PADV 4</td>
<td></td>
<td>This item is already validated and ready for dispense</td>
</tr>
<tr>
<td>MTP 5</td>
<td>PRE Item 2-3</td>
<td></td>
<td></td>
<td>This item is not validated yet</td>
</tr>
<tr>
<td>MTP 6</td>
<td>PRE 3</td>
<td>PRE Item 3-1</td>
<td></td>
<td>This item is not validated yet</td>
</tr>
</tbody>
</table>

Note that Medication Treatment Plan Items (documents) are optional and may not be present. They are included in the examples for describing the situation where a Medication Treatment Plan Planner is being used.

**Example 1: Standard query**

Used Query Parameters:
- Patient ID
- Document Status

This is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prescriptions</td>
<td>Related documents</td>
</tr>
<tr>
<td>PRE 2</td>
<td>MTP 3, MTP 4, MTP 5, PADV 3, PADV 4, DIS 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prescriptions</th>
<th>Related documents</th>
<th>PRE Item 3-1 of PRE 3 is not validated yet and therefore PRE 3 shall be returned as result. MTP 6 is related to PRE 3 and shall also be returned as a result.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRE 3</td>
<td>MTP 6</td>
<td></td>
</tr>
</tbody>
</table>
Example 2: Search for a specific prescription

Query Parameters set:

- Patient ID
- Document Status
- Document uniqueId of the specific prescription (e.g., because patient showed a paper prescription with the uniqueId printed on it)

In case the uniqueId of PRE 2 is given as query parameter, this is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRE 2</td>
<td>PRE Item 2-2 of PRE 2 is not validated yet and therefore PRE 2 shall be returned as result. PADV 3, PADV 4 and DIS 2 are all documents which are related to (some PRE Items on) PRE 2 and shall also be returned as result.</td>
</tr>
<tr>
<td>MTP 3, MTP 4, MTP 5, PADV 3, PADV 4, DIS 2</td>
<td></td>
</tr>
</tbody>
</table>

Note: In case the uniqueId of PRE 1 is given as query parameter, it would result in an empty result set, because PRE 1 would be the only possible return but contains no Prescription Item which is ready to be validated.

3.1.4.1.2.1.1.4 FindPrescriptionsForDispense

Find prescriptions and their related documents (XDSDocumentEntry objects) containing Prescription Items ready to be dispensed for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

Returns: XDSDocumentEntry objects according to the following business rules:

- Business rule 1: Returns Prescription documents matching the query parameters:
  - XDSDocumentEntry matches all required query parameters (PatientID, Status)
  - XDSDocumentEntry matches all other optional query parameters
  - FormatCode matches urn:ihe:pharm:pre:2010
  - Prescription document contains at least one Prescription Item ready to dispense
• Scenario 1 “Including validation step”: A Prescription Item is ready to dispense if the last Pharmaceutical Advice Item related to it has statusCode set to “completed” and Observation Code set to either OK or CHANGE.\(^{31}\)

• Scenario 2 “Not including validation step”: A Prescription Item is ready to be dispensed, if either no Pharmaceutical Advice Item related to it exists or the last Pharmaceutical Advice Item related to it has statusCode set to “completed” and Observation Code set to either OK or CHANGE.\(^{32}\)

Business rule 2: Returns related Medication Treatment Plan documents to the Prescriptions found if Medication Treatment Plan Planner is used

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches \texttt{urn:ihe:pharm:mtp:2015}
- Medication Treatment Plan document contains a Medication Treatment Plan Entry Item related to a Prescription Item of the found Prescription documents.\(^{33}\)

Business rule 5: Returns related Dispense documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches \texttt{urn:ihe:pharm:dis:2010}
- Dispense document contains a Dispense Entry Item related to a Prescription Item of the found Prescription documents.\(^{34}\)

Business rule 4: Returns related Pharmaceutical Advice documents to the Prescriptions found

- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches \texttt{urn:ihe:pharm:padv:2010}
- Pharmaceutical Advice\(^{35}\) document contains

\(^{31}\) See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the statusCode and Observation code elements

\(^{32}\) See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details about the statusCode and Observation code elements

\(^{33}\) See the Pharmacy Prescription Content Profile (MTP) for details how the relation between Medication Treatment Plan Entries and Prescription Items is defined, chapter: “Prescription Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.2)

\(^{34}\) See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Prescription Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)
(1) Pharmaceutical Advice document contains a Pharmaceutical Advice Entry Item related to a Prescription Item of the found Prescription documents

(2) Pharmaceutical Advice document contains a Pharmaceutical Advice Entry Item related to a Medication Treatment Plan Item of the found Medication Treatment Plans documents

**Explanation**

Returning Prescription documents according to business rule 1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Medication Treatment Plan, Pharmaceutical Advice and Dispense documents to the Prescriptions (Business rule 2 to 4), is dependent on the primary result (found Prescriptions) only and contains just directly related documents.

**Query parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Attribute</th>
<th>Opt</th>
<th>Mult</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XDSDocumentEntryPatientId</td>
<td>XDSDocumentEntry. patientId</td>
<td>R</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryEntryUUID</td>
<td>XDSDocumentEntry. entryUUID</td>
<td>O^1</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryUniqueId</td>
<td>XDSDocumentEntry. uniqueId</td>
<td>O^1</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryPracticeSettingCode^3</td>
<td>XDSDocumentEntry. practiceSettingCode</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryCreationTimeFrom</td>
<td>Lower value of XDSDocumentEntry. creationTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryCreationTimeTo</td>
<td>Upper value of XDSDocumentEntry. creationTime</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStartTimeFrom</td>
<td>Lower value of XDSDocumentEntry. serviceStart</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStartTimeTo</td>
<td>Upper value of XDSDocumentEntry. serviceStart</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStopTimeFrom</td>
<td>Lower value of XDSDocumentEntry. serviceStop</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStopTimeTo</td>
<td>Upper value of XDSDocumentEntry. serviceStop</td>
<td>O</td>
<td>--</td>
</tr>
</tbody>
</table>

^3 See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Medication Treatment Plan-/Prescription Items is defined, chapter: “Pharmaceutical Advice Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.3)
### Example Parameters of the FindPrescriptionsForDispense Query

#### Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Attribute</th>
<th>Opt</th>
<th>Mult</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XDSDocumentEntry.serviceStopTime</td>
<td>XDSDocumentEntry.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$XDSDocumentEntryHealthcareFacilityTypeCode³</td>
<td>XDSDocumentEntry.</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryEventCodeList³</td>
<td>XDSDocumentEntry.</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryConfidentialityCode³</td>
<td>XDSDocumentEntry.</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryAuthorPerson⁴</td>
<td>XDSDocumentEntry.</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryStatus</td>
<td>XDSDocumentEntry.</td>
<td>R</td>
<td>M</td>
</tr>
</tbody>
</table>

1. Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueIds shall be specified. This transaction shall return an error if both parameters are specified.

3. Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

4. The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute).

#### Examples for the “FindPrescriptionsForDispense” query

Assume the following situation of persisted documents in the Prescription-/Pharmaceutical Advice- and Dispense repositories:

<table>
<thead>
<tr>
<th>Treatment Plans</th>
<th>Prescriptions</th>
<th>Pharmaceutical Advice</th>
<th>Dispenses</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTP 1</td>
<td>PRE 1</td>
<td>PRE Item 1-1</td>
<td>PADV 1</td>
<td>DIS 1</td>
</tr>
<tr>
<td>MTP 2</td>
<td>PRE Item 1-2</td>
<td>PADV 2</td>
<td>DIS 2</td>
<td>This item is already validated and dispensed</td>
</tr>
<tr>
<td>MTP 3</td>
<td>PRE 2</td>
<td>PRE Item 2-1</td>
<td>PADV 3</td>
<td>DIS 2</td>
</tr>
<tr>
<td>MTP 4</td>
<td>PRE Item 2-2</td>
<td>PADV 4</td>
<td></td>
<td>This item is already validated and ready for dispense</td>
</tr>
<tr>
<td>MTP 5</td>
<td>PRE Item 2-3</td>
<td></td>
<td></td>
<td>This item is not validated yet</td>
</tr>
<tr>
<td>MTP 6</td>
<td>PRE 3</td>
<td>PRE Item 3-1</td>
<td></td>
<td>This item is not validated yet</td>
</tr>
</tbody>
</table>

Note that Medication Treatment Plan Items (documents) are optional and may not be present. They are included in the examples for describing the situation where a Medication Treatment Plan Planner is being used.
Example 1: Standard query

Used Query Parameters:

- Patient ID
- Document Status

This is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prescriptions</td>
<td>Related documents</td>
</tr>
<tr>
<td>PRE 1</td>
<td>MTP 1, MTP 2, PADV 1, PADV 2, DIS 1</td>
</tr>
<tr>
<td></td>
<td>PRE Item 1-2 of PRE 1 is validated but not dispensed yet, therefore PRE 1 shall be returned as result. MTP 1, MTP 2, PADV 1, PADV 2 and DIS 1 are all documents which are related to (some PRE Items on) PRE 1 and shall also be returned as result.</td>
</tr>
<tr>
<td>PRE 2</td>
<td>MTP 3, MTP 4, MTP 5, PADV 3, PADV 4, DIS 2</td>
</tr>
<tr>
<td></td>
<td>PRE Item 2-2 of PRE 2 is validated but not dispensed yet and therefore PRE 2 shall be returned as result. MTP 3, MTP 4, MTP 5, PADV 3, PADV 4 and DIS 2 are all documents which are related to (some PRE Items on) PRE 2 and shall also be returned as result.</td>
</tr>
</tbody>
</table>

Example 2: Search for a specific prescription

Query Parameters set:

- Patient ID
- Document Status
- Document uniqueld of the specific prescription (e.g., because patient showed a paper prescription with the uniqueld printed on it)

In case the uniqueld of PRE 1 is given as query parameter, this is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prescriptions</td>
<td>Related documents</td>
</tr>
<tr>
<td>PRE 1</td>
<td>MTP 1, MTP 2, PADV 1, PADV 2, DIS 1</td>
</tr>
<tr>
<td></td>
<td>PRE Item 1-2 of PRE 1 is validated but not dispensed yet, therefore PRE 1 shall be returned as result. MTP 1, MTP 2, PADV 1, PADV 2 and DIS 1 are all documents which are related to (some PRE Items on) PRE 1 and shall also be returned as result.</td>
</tr>
</tbody>
</table>

Note: In case the uniqueld of PRE 3 is given as query parameter, it would result in an empty result set, because PRE 3 would be the only possible return but contains no Prescription Item which is ready to be dispensed.
3.1.4.1.2.1.1.5 FindMedicationList

Find the Medication List On-Demand Document (XDSDocumentEntry object) according to the Pharmacy Medication List (PML) Profile containing Prescription- and Dispense Items for a given patientID and other matching attributes.

**Returns:** XDSDocumentEntry object according to the following business rules.

Business rule 1: Returns *Medication List* documents

This is the basic and ready to implement mechanism to retrieve a medication list. In this case, the Community Pharmacy Manager contains, or has access to, business rules to retrieve an existing list, or to create the patient’s medication lists from the Pharmacy documents.

A set of parameters is provided in the query to obtain such list:

- XDSDocumentEntry matches all the required query parameters below (PatientID, …)
- FormatCode of the returned document match urn:ihe:pharm:pml:2013 (implicit business rule, parameter shall not be given in parameter list)

**Query parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Attribute</th>
<th>Opt</th>
<th>Mult</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XDSDocumentEntryPatientId</td>
<td>XDSDocumentEntry.patientId</td>
<td>R</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStartFrom</td>
<td></td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceStartTo</td>
<td></td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceEndFrom</td>
<td></td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryServiceEndTo</td>
<td></td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryFormatCode</td>
<td></td>
<td>O(^{36})</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryType</td>
<td>XDSDocumentEntry.objectType</td>
<td>R</td>
<td>M</td>
</tr>
</tbody>
</table>

\(^{36}\) Note: Omitting this parameter means that no filtering according to format code takes place, so “all” available data types will be returned.

\(^{37}\) Note: Omitting this parameter means that “all” available types of Medication List documents (on-demand created or previously persisted snapshots) are returned.

\(^{38}\) See ITI TF-2a:3.18.4.1.2.3.6.2
Explanation of yellow parameters

1) $XDSDocumentEntryServiceStartFrom, $XDSDocumentEntryServiceStartTo

This query parameter is used to find all medication treatments that were started during the interval specified by the requester. The exact definition of the starting point of a medication treatment is not in the scope of this profile.

For example:

(1) find and return all medication treatments that were started between Nov 2012 to June 2013
(2) find and return all medication treatments that were started in the last 3 months

2) $XDSDocumentEntryServiceEndFrom, $XDSDocumentEntryServiceEndTo,

This query parameter is used to find all medication treatments that were finished / completed in the interval specified by the requester. The exact definition of the point where a medication treatment is finished/completed is not in the scope of this profile.

For example:

(1) find and return all medication treatments that were completed between Nov 2012 to June 2013
(2) find and return all medication treatments that were completed in the last 3 months or are not yet completed

3) $XDSDocumentEntryFormatCode

If this parameter is given just the given type of information shall be returned in the Medication List.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:ihe:pharm:mtp:2015</td>
<td>Medication Treatment Plan Items shall be returned (and optional the related Pharmaceutical Advice documents related to them).</td>
</tr>
<tr>
<td>urn:ihe:pharm:pre:2010</td>
<td>Prescription Items shall be returned (and optional the related Pharmaceutical Advice documents related to them).</td>
</tr>
<tr>
<td>urn:ihe:pharm:dis:2010</td>
<td>Dispense Items shall be returned (and optional the related Pharmaceutical Advice documents related to them).</td>
</tr>
</tbody>
</table>

4) $XDSDocumentEntryType

If this parameter is given documents of just the provided document entry type (on-demand or stable) shall be returned.
### 3.1.4.1.2.1.1.6 FindMedicationTreatmentPlans

Find planned medications and their related documents (XDSDocumentEntry objects) containing Medication Treatment Plan Items for a given patientID and other matching attributes. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

**Returns:** XDSDocumentEntry objects according to the following business rules.

**Business rule 1:** Returns *Medication Treatment Plan* documents matching the query parameters:
- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- XDSDocumentEntry matches all other optional query parameters

**Business rule 2:** Returns related *Prescription* documents to the Medication Treatment Plans found
- XDSDocumentEntry matches all required query parameters (PatientID, Status)
- FormatCode matches `urn:ihe:pharm:pre:2010`
- Prescription document contains a Prescription Entry Item related to a Medication Treatment Plan Item of the found Medication Treatment Plan documents.\(^{40}\)

**Business rule 3:** Returns related *Dispense* documents to the Medication Treatment Plans found
- XDSDocumentEntry matches all required query parameters (PatientID, Status)

---

\(^{39}\) Note: This parameter is applicable only if the “Persistence of Retrieved Documents” Option is supported.

\(^{40}\) See the Pharmacy Prescription Content Profile (PRE) for details how the relation between Prescription Entries and Medication Treatment Plan Items is defined, chapter: “Prescription Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.2)
• Dispense document contains a Dispense Entry Item related to a Medication Treatment Plan Item of the found Medication Treatment Plan documents.41

Business rule 4: Returns related Pharmaceutical Advice documents to the Medication Treatment Plans found

• XDSDocumentEntry matches all required query parameters (PatientID, Status)
• FormatCode matches urn:ihe:pharm:padv:2010
• Pharmaceutical Advice42 document contains
  • (1) a Pharmaceutical Advice Entry Item related to a Medication Treatment Plan Item of the found Medication Treatment Plan documents
  • (2) a Pharmaceutical Advice Entry Item related to a Prescription Item of the found Medication Treatment Plan documents
  • (3) a Pharmaceutical Advice Entry Item related to a Dispense Item of the found Dispense documents.

Explanation
Returning Medication Treatment Plan documents according to business rule 1 is the primary result of the query, where all optional query parameters which might affect the result of the query are applied.

The secondary result of the query, the related Pharmaceutical Advice, Prescription and Dispense documents to the Medication Treatment Plans (Business rules 2 to 4), is dependent on the primary result (found Medication Treatment Plans) only and contains just directly related documents.

Query parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Attribute</th>
<th>Opt</th>
<th>Mult</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XDSDocumentEntryPatientId</td>
<td>XDSDocumentEntry. patientId</td>
<td>R</td>
<td>--</td>
</tr>
<tr>
<td>$XDSDocumentEntryEntryUUID</td>
<td>XDSDocumentEntry. entryUUID</td>
<td>O^1</td>
<td>M</td>
</tr>
<tr>
<td>$XDSDocumentEntryUniqueId</td>
<td>XDSDocumentEntry. uniqueId</td>
<td>O^1</td>
<td>M</td>
</tr>
</tbody>
</table>

41 See the Pharmacy Dispense Content Profile (DIS) for details how the relation between Dispense Entries and Medication Treatment Plan Items is defined, chapter: “Dispense Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.4)

42 See the Pharmacy Pharmaceutical Advice Content Profile (PADV) for details how the relation between Pharmaceutical Advice Entries and Medication Treatment Plan-/Prescription-/Dispense Items is defined, chapter: “Pharmaceutical Advice Item Entry Content Module” (1.3.6.1.4.1.19376.1.9.1.3.3)
### Parameter Name | Attribute | Opt | Mult
--- | --- | --- | ---
$XDSDocumentEntryPracticeSettingCode$ | XDSDocumentEntry. practiceSettingCode | O | M
$XDSDocumentEntryCreationTimeFrom$ | Lower value of XDSDocumentEntry. creationTime | O | --
$XDSDocumentEntryCreationTimeTo$ | Upper value of XDSDocumentEntry. creationTime | O | --
$XDSDocumentEntryServiceStartTimeFrom$ | Lower value of XDSDocumentEntry. serviceStartTime | O | --
$XDSDocumentEntryServiceStartTimeTo$ | Upper value of XDSDocumentEntry. serviceStartTime | O | --
$XDSDocumentEntryServiceStopTimeFrom$ | Lower value of XDSDocumentEntry. serviceStopTime | O | --
$XDSDocumentEntryServiceStopTimeTo$ | Upper value of XDSDocumentEntry. serviceStopTime | O | --
$XDSDocumentEntryHealthcareFacilityTypeCode$ | XDSDocumentEntry. healthcareFacilityTypeCode | O | M
$XDSDocumentEntryEventCodeList$ | XDSDocumentEntry. eventCodeList | O | M
$XDSDocumentEntryConfidentialityCode$ | XDSDocumentEntry. confidentialityCode | O | M
$XDSDocumentEntryAuthorPerson$ | XDSDocumentEntry. Author | O | M
$XDSDocumentEntryStatus$ | XDSDocumentEntry. Status | R | M

1Either $XDSDocumentEntryEntryUUID$ or $XDSDocumentEntryUniqueId$ shall be specified. This transaction shall return an error if both parameters are specified.

3Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

4The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute)

**Examples for the “FindMedicationTreatmentPlans” query**

Assume the following situation of persisted documents in the Medication Treatment Plan-/Pharmaceutical Advice- /Prescription- and Dispense repositories:

1520
### Example 1: Standard query

Used Query Parameters:
- **Patient ID**: 1525
- **Document Status**: 1525

This is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Medication Treatment Plans</strong></td>
<td><strong>Related documents</strong></td>
</tr>
<tr>
<td>MTP 1</td>
<td>PRE 1, PADV 1, DIS 1</td>
</tr>
<tr>
<td>MTP 2</td>
<td>PRE 1, PADV 2</td>
</tr>
<tr>
<td>MTP 3</td>
<td></td>
</tr>
<tr>
<td>MTP 4</td>
<td>PRE 2, PADV 3, PADV 4, PADV 5, DIS 2</td>
</tr>
<tr>
<td>MTP 5</td>
<td>PRE 3, DIS 3</td>
</tr>
</tbody>
</table>
Example 2: Search for a specific medication treatment plan

Query Parameters set:
- Patient ID
- Document Status
- Document uniqueld of the specific medication treatment plan

In case the uniqueld of MTP 2 is given as query parameter, this is what should be returned by the query:

<table>
<thead>
<tr>
<th>Returned XDSDocumentEntries</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prescriptions</strong>&lt;br&gt;MTP 2</td>
<td><strong>Related documents</strong>&lt;br&gt;PRE 1, PADV 2&lt;br&gt;PADV 1, DIS 1</td>
</tr>
</tbody>
</table>

3.1.4.1.2.2 Stored Query IDs

The Registry Stored Query (ITI-18) transaction defines several kinds of Stored Queries (FindDocuments, FindSubmissionSets, etc.).

The PHARM-1 transaction has to provide a different set of Stored Queries. They are assigned the following Query IDs. These IDs are used in the AdhocQueryRequest to reference queries stored on the Community Pharmacy Manager Actor. Query IDs are in UUID format (RFC4122). An error shall be returned when an unsupported stored query ID is received.

<table>
<thead>
<tr>
<th>Query Name</th>
<th>Query ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>FindPrescriptions</td>
<td>urn:uuid:0e6095c5-dc3d-47d9-a219-047064086d92</td>
</tr>
<tr>
<td>FindDispenses</td>
<td>urn:uuid:ac79c7c7-f21b-4c88-ab81-57e4889e8758</td>
</tr>
<tr>
<td>FindPrescriptionsForValidation</td>
<td>urn:uuid:ca43b20-0254-102e-8469-a6af440562e8</td>
</tr>
<tr>
<td>FindPrescriptionsForDispense</td>
<td>urn:uuid:c875eb9c-0254-102e-8469-a6af440562e8</td>
</tr>
<tr>
<td>FindMedicationList</td>
<td>urn:uuid:80ebbd83-53c1-4453-9860-349585962af6</td>
</tr>
<tr>
<td>FindMedicationTreatmentPlans</td>
<td>urn:uuid:c85f5ade-81c1-44b6-8f7c-48b9cd6b9489</td>
</tr>
</tbody>
</table>
3.1.4.1.2.3 Web Services Transport

The Registry Stored Query (ITI-18) transaction defines the transmission using Web Services.

This chapter describes the differences of the PHARM-1 transaction to the ITI-18.

IHE-WSP201) The attribute /wsdl:definitions/@name shall be “CommunityPharmacyManager”.

The following WSDL naming conventions shall apply:

wsdl:definitions/@name="CommunityPharmacyManager":
query message -> "QueryPharmacyDocuments_Message"
query response -> "QueryPharmacyDocuments_Response_Message"
portType -> "CommunityPharmacyManager_ProtType"
operation -> "QueryPharmacyDocuments"
SOAP 1.2 binding -> "CommunityPharmacyManager_Binding_Soap12"
SOAP 1.2 port -> "CommunityPharmacyManager_Port_Soap12"

IHE-WSP202) The targetNamespace of the WSDL shall be “urn:ihe:iti:xds-b:2007”

These are the requirements for the Registry Stored Query transaction presented in the order in which they would appear in the WSDL definition:

- The following types shall be imported (xsd:import) in the /definitions/types section:
  namespace=" urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0", schemaLocation="query.xsd"
- The /definitions/message/part/@element attribute of the Find Prescriptions Request message shall be defined as “query:AdhocQueryRequest”
- The /definitions/message/part/@element attribute of the Find Prescriptions Response message shall be defined as “query:AdhocQueryResponse”
- The /definitions/portType/operation/input/@wsaw:Action attribute for the Find Prescriptions Request message shall be defined as “urn:ihe:pharm:cmpd:2010:QueryPharmacyDocuments”
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Find Prescriptions Response message shall be defined as “urn:ihe:pharm:cmpd:2010:QueryPharmacyDocumentsResponse”
- The /definitions/binding/operation/soap12:operation/@soapAction attribute should be defined as “urn:ihe:pharm:cmpd:2010:QueryPharmacyDocuments"

The following WSDL fragment shows an example of Find Prescription transaction definition:

```xml
<?xml version="1.0" encoding="utf-8"?>
<definitions ...
...>
<types>
  <xsd:schema elementFormDefault="qualified" targetNamespace="urn:ihe:iti:xds-b:2007">
```

```xml
..."query.xsd"
```

```xml
..."query:AdhocQueryRequest"
```

```xml
..."query:AdhocQueryResponse"
```

```xml
..."urn:ihe:pharm:cmpd:2010:QueryPharmacyDocuments"
```

```xml
..."urn:ihe:pharm:cmpd:2010:QueryPharmacyDocumentsResponse"
```

```xml
..."urn:ihe:pharm:cmpd:2010:QueryPharmacyDocuments"
```
3.1.4.1.2.3.1 Sample SOAP Messages

The samples in the following two sections show a typical SOAP request and its relative SOAP response.

3.1.4.1.2.3.1.1 Sample Query Prescription SOAP Request

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing">
    <s:Header>
        <a:MessageID>urn:uuid:dfef19ad-dc13-49c1-a3c7-e3742531f9b3</a:MessageID>
        <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
        <a:ReplyTo><a:Address>http://localhost/service/CommunityPharmacyManager.svc</a:Address></a:ReplyTo>
    </s:Header>
    <s:Body>
        <query:AdhocQueryRequest>
            see Sample Query Request below
        </query:AdhocQueryRequest>
    </s:Body>
</s:Envelope>

3.1.4.1.2.3.1.2 Sample Query Prescription SOAP Response

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:a="http://www.w3.org/2005/08/addressing">
    <s:Header>
    </s:Header>
    <s:Body>
    </s:Body>
</s:Envelope>
3.1.4.1.3 Expected Actions

The Community Pharmacy Manager Actor shall do the same actions as described for the Document Registry Actor in the Registry Stored Query (ITI-18) transaction.

3.1.4.1.3.1 Sample Query Request

This example query specifies:

- The FindPrescriptionsForValidation query (id attribute of AdhocQuery element)
- patientID st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO
- Return Approved documents only
- Time range (creation time) 200412252300 to 200501010800

Note that ebRS 3.0 specifies the use of Slot to specify name/value(s) pairs as parameters to a Stored Query.

Note: query parameter names are highlighted for readability.
3.1.4.1.3.2 Sample Query Response

This sample query response corresponds to the above query. Note that the query response message is coded in version 3.0 ebRIM and ebRS. This sample response and the ebXML Registry version 3.0 schema files are available online. The Implementation Guide found at http://wiki.ihe.net/index.php?title=ITI_Implementation_Guide contains such supplemental material.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<AdhocQueryResponse
 xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
 xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:schemaLocation="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0 file:/query.xsd"
 status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
  <rim:RegistryObjectList>
    <rim:ExtrinsicObject
     xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
     xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
     id="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
     isOpaque="false"
     mimeType="text/xml"
     objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
     status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Approved">
      <rim:Slot name="URI">
        <rim:Value>http://localhost:8080/XDS/Repository/08a15a6f-5b4a-42de-8f95-89474f83abdf.xml</rim:Value>
      </rim:Slot>
      <rim:Slot name="authorInstitution">
        <rim:Value>Some Hospital^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45</rim:Value>
      </rim:Slot>
      <rim:Slot name="creationTime">
        <rim:Value>200412261119</rim:Value>
      </rim:Slot>
      <rim:Slot name="hash">
        <rim:Value>4cf4f82d78b5e2aac35c31bca8cb79fe6bd6a41e</rim:Value>
      </rim:Slot>
    </rim:ExtrinsicObject>
  </rim:RegistryObjectList>
</AdhocQueryResponse>
```
<table>
  <tr>
    <td>1740</td>
    <td><rim:Slot name="languageCode">
        <rim:ValueList>
          <rim:Value>en-us</rim:Value>
        </rim:ValueList>
      </rim:Slot></td>
  </tr>
  <tr>
    <td>1745</td>
    <td><rim:Slot name="serviceStartTime">
        <rim:ValueList>
          <rim:Value>200412230800</rim:Value>
        </rim:ValueList>
      </rim:Slot></td>
  </tr>
  <tr>
    <td>1750</td>
    <td><rim:Slot name="serviceStopTime">
        <rim:ValueList>
          <rim:Value>200412230801</rim:Value>
        </rim:ValueList>
      </rim:Slot></td>
  </tr>
  <tr>
    <td>1755</td>
    <td><rim:Slot name="size">
        <rim:ValueList>
          <rim:Value>54449</rim:Value>
        </rim:ValueList>
      </rim:Slot></td>
  </tr>
  <tr>
    <td>1760</td>
    <td><rim:Slot name="sourcePatientId">
        <rim:ValueList>
          <rim:Value>jd12323^^^wsh</rim:Value>
        </rim:ValueList>
      </rim:Slot></td>
  </tr>
  <tr>
    <td>1765</td>
    <td><rim:Slot name="sourcePatientInfo">
        <rim:ValueList>
          <rim:Value>PID-3|pid1^^^domain</rim:Value>
          <rim:Value>PID-5|Doe^^^John</rim:Value>
          <rim:Value>PID-7|19560527</rim:Value>
          <rim:Value>PID-8|M</rim:Value>
          <rim:Value>PID-11|100 Main St^^Metropolis^Il^44130^USA</rim:Value>
        </rim:ValueList>
      </rim:Slot></td>
  </tr>
  <tr>
    <td>1770</td>
    <td><rim:Slot name="Name">
        <rim:LocalizedString charset="UTF-8" value="Pharmacy Prescription" xml:lang="en-us" />
      </rim:Slot></td>
  </tr>
  <tr>
    <td>1775</td>
    <td><rim:Slot name="Classification">
        <classificationScheme>urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a</classificationScheme>
        <classifiedObject>urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf</classifiedObject>
        <id>urn:uuid:ac872fc0-1c6e-439f-84d1-f76770a0ccdf</id>
        <nodeRepresentation>57833-6</nodeRepresentation>
        <objectType>urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification</objectType>
      </rim:Slot></td>
  </tr>
  <tr>
    <td>1780</td>
    <td><rim:Slot name="codingScheme">
        <classificationScheme>urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f</classificationScheme>
        <classifiedObject>urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf</classifiedObject>
      </rim:Slot></td>
  </tr>
</table>
id="urn:uuid:f1a8c8e4-3593-4777-b7e0-8b0773378705"
nodeRepresentation="N"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>Connect-a-thon confidentialityCodes</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Normal" xml:lang="en-us"/>
  </rim:Name>
  <rim:Description/>
</rim:Classification>
</rim:Classification>
classificationScheme="urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"
classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
id="urn:uuid:b6e49c73-96c8-4058-8c95-914d83bd262a"
nodeRepresentation="urn:ihe:pharm:pre:2010"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>Connect-a-thon formatCodes</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Pharmacy Prescription" xml:lang="en-us"/>
  </rim:Name>
  <rim:Description/>
</rim:Classification>
</rim:Classification>
classificationScheme="urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"
classifiedObject="urn:uuid:61e2b376-d74a-4984-ac21-dcd0b8890f9d"
nodeRepresentation="Emergency Department"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>Connect-a-thon healthcareFacilityTypeCodes</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Emergency Department" xml:lang="en-us"/>
  </rim:Name>
  <rim:Description/>
</rim:Classification>
</rim:Classification>
classificationScheme="urn:uuid:ccc55f98-8b07-4b77-a05e-ae952c785ead"
classifiedObject="urn:uuid:fb7677c5-4c2f-485d-9010-dce0f3cd4ad5"
nodeRepresentation="Cardiology"
objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>Connect-a-thon practiceSettingCodes</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Cardiology" xml:lang="en-us"/>
  </rim:Name>
  <rim:Description/>
</rim:Classification>
</rim:Classification>
### 3.1.5 Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the XDS Security Considerations Section (see ITI TF-1: 10.7).
3.1.5.1 Security Audit Considerations
The Actors involved shall record audit events according to the following:

### 3.1.5.1.1 Querying actor audit message:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Opt</th>
<th>Value Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventAuditMessage/EventIdentification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventID</td>
<td>M</td>
<td>EV(110112, DCM, “Query”)</td>
</tr>
<tr>
<td>EventActionCode</td>
<td>M</td>
<td>“E” (Execute)</td>
</tr>
<tr>
<td>EventDateTime</td>
<td>M</td>
<td>not specialized</td>
</tr>
<tr>
<td>EventOutcomeIndicator</td>
<td>M</td>
<td>not specialized</td>
</tr>
<tr>
<td>EventTypeCode</td>
<td>M</td>
<td>EV(“PHARM-1”, “IHE Transactions”, “Query Pharmacy Documents”)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source (Document Consumer)</th>
<th>(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Requestor (0..n)</td>
<td></td>
</tr>
<tr>
<td>Destination (Document Registry)</td>
<td>(1)</td>
</tr>
<tr>
<td>Audit Source (Document Consumer)</td>
<td>(1)</td>
</tr>
<tr>
<td>Patient (0..1)</td>
<td></td>
</tr>
<tr>
<td>Query Parameters(1)</td>
<td></td>
</tr>
</tbody>
</table>

Where:

| Source (AuditMessage/ActiveParticipant) | | | |
|----------------------------------------|-----------------------------|-----------------------------------------------------------------|
| UserID                                 | M                           | The content of the <wsa:ReplyTo/> element.                      |
| AlternativeUserID                      | M                           | the process ID as used within the local operating system in the local system logs. |
| UserName                               | U                           | not specialized                                                 |
| UserIsRequestor                        | M                           | “true”                                                          |
| RoleIDCode                             | M                           | EV(110153, DCM, “Source”)                                       |
| NetworkAccessPointTypeCode             | M                           | “1” for machine (DNS) name, “2” for IP address                 |
| NetworkAccessPointID                   | M                           | The machine name or IP address, as specified in RFC 3881.       |

| Human Requestor (if known) (AuditMessage/ActiveParticipant) | | | |
|----------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------|
| UserID                                                             | M                           | Identity of the human that initiated the transaction.           |
| AlternativeUserID                                                  | U                           | not specialized                                                |
| UserName                                                           | U                           | not specialized                                                |
| UserIsRequestor                                                    | M                           | “true”                                                          |
| RoleIDCode                                                         | U                           | Access Control role(s) the user holds that allows this transaction. |
| NetworkAccessPointTypeCode                                         | NA                          |                                                                  |
| NetworkAccessPointID                                               | NA                          |                                                                  |
### Destination

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>M</td>
<td>SOAP endpoint URI.</td>
</tr>
<tr>
<td>AlternativeUserID</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>UserName</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>UserIsRequestor</td>
<td>M</td>
<td>“false”</td>
</tr>
<tr>
<td>RoleIDCode</td>
<td>M</td>
<td>EV(110152, DCM, “Destination”)</td>
</tr>
<tr>
<td>NetworkAccessPointTypeCode</td>
<td>M</td>
<td>“1” for machine (DNS) name, “2” for IP address</td>
</tr>
<tr>
<td>NetworkAccessPointID</td>
<td>M</td>
<td>The machine name or IP address, as specified in RFC 3881.</td>
</tr>
</tbody>
</table>

### Audit Source

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuditSourceID</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>AuditEnterpriseSiteID</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>AuditSourceTypeCode</td>
<td>U</td>
<td>not specialized</td>
</tr>
</tbody>
</table>

### Patient

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParticipantObjectTypeCode</td>
<td>M</td>
<td>“1” (Person)</td>
</tr>
<tr>
<td>ParticipantObjectTypeCodeRole</td>
<td>M</td>
<td>“1” (Patient)</td>
</tr>
<tr>
<td>ParticipantObjectDataLifeCycle</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectIDTypeCode</td>
<td>M</td>
<td>EV(2, RFC-3881, “Patient Number”)</td>
</tr>
<tr>
<td>ParticipantObjectSensitivity</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectID</td>
<td>M</td>
<td>The patient ID in HL7 CX format.</td>
</tr>
<tr>
<td>ParticipantObjectName</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectQuery</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectDetail</td>
<td>U</td>
<td>not specialized</td>
</tr>
</tbody>
</table>

### Query Parameters

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParticipantObjectTypeCode</td>
<td>M</td>
<td>“2” (system object)</td>
</tr>
<tr>
<td>ParticipantObjectTypeCodeRole</td>
<td>M</td>
<td>“24” (query)</td>
</tr>
<tr>
<td>ParticipantObjectDataLifeCycle</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectIDTypeCode</td>
<td>M</td>
<td>EV(“PHARM-1”, “IHE Transactions”, “Query Pharmacy Documents”)</td>
</tr>
<tr>
<td>ParticipantObjectSensitivity</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectID</td>
<td>M</td>
<td>Stored Query ID (UUID)</td>
</tr>
<tr>
<td>ParticipantObjectName</td>
<td>C</td>
<td>If known the value of <a href="">ihe:HomeCommunityId</a>/</td>
</tr>
<tr>
<td>ParticipantObjectQuery</td>
<td>M</td>
<td>the AdhocQueryRequest, base64 encoded.</td>
</tr>
</tbody>
</table>
ParticipantObjectDetail C
The ParticipantObjectDetail element may occur more than once.
In one element, set “QueryEncoding” as the value of the attribute type. Set the attribute value to the character encoding, such as “UTF-8”, used to encode the ParticipantObjectQuery before base64 encoding.
In another element, set “urn:ihe:iti:xca:2010:homeCommunityId” as the value of the attribute type and the value of the homeCommunityID as the value of the attribute value, if known.

3.1.5.1.2 Community Pharmacy Manager audit message:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Opt</th>
<th>Value Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventID</td>
<td>M</td>
<td>EV(110112, DCM, “Query”)</td>
</tr>
<tr>
<td>EventActionCode</td>
<td>M</td>
<td>“E” (Execute)</td>
</tr>
<tr>
<td>EventDateTime</td>
<td>M</td>
<td>not specialized</td>
</tr>
<tr>
<td>EventOutcomeIndicator</td>
<td>M</td>
<td>not specialized</td>
</tr>
<tr>
<td>EventTypeCode</td>
<td>M</td>
<td>EV(“PHARM-1”, “IHE Transactions”, “Query Pharmacy Documents”)</td>
</tr>
</tbody>
</table>

Source (Document Consumer) (1)

Destination (Document Registry) (1)

Audit Source (Document Registry) (1)

Patient (0..1)

Query Parameters(1)

Where:

<table>
<thead>
<tr>
<th>Source (Document Consumer)</th>
<th>Opt</th>
<th>Value Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>M</td>
<td>The content of the <a href="">wsa:ReplyTo/</a> element.</td>
</tr>
<tr>
<td>AlternativeUserID</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>UserName</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>UserIsRequestor</td>
<td>M</td>
<td>“true”</td>
</tr>
<tr>
<td>RoleIDCode</td>
<td>M</td>
<td>EV(110153, DCM, “Source”)</td>
</tr>
<tr>
<td>NetworkAccessPointTypeCode</td>
<td>M</td>
<td>“1” for machine (DNS) name, “2” for IP address</td>
</tr>
<tr>
<td>NetworkAccessPointID</td>
<td>M</td>
<td>The machine name or IP address, as specified in RFC 3881.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination (Document Registry)</th>
<th>Opt</th>
<th>Value Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>M</td>
<td>SOAP endpoint URI.</td>
</tr>
<tr>
<td>AlternativeUserID</td>
<td>M</td>
<td>the process ID as used within the local operating system in the local system logs.</td>
</tr>
<tr>
<td>UserName</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>UserIsRequestor</td>
<td>M</td>
<td>“false”</td>
</tr>
<tr>
<td>RoleIDCode</td>
<td>M</td>
<td>EV(110152, DCM, “Destination”)</td>
</tr>
<tr>
<td>NetworkAccessPointTypeCode</td>
<td>M</td>
<td>“1” for machine (DNS) name, “2” for IP address</td>
</tr>
<tr>
<td><strong>NetworkAccessPointID</strong></td>
<td>M</td>
<td>The machine name or IP address, as specified in RFC 3881.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---</td>
<td>--------------------------------------------------------</td>
</tr>
</tbody>
</table>

### AuditSource

<table>
<thead>
<tr>
<th><strong>AuditSourceID</strong></th>
<th>U</th>
<th>not specialized</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuditEnterpriseSiteID</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>AuditSourceTypeCode</td>
<td>U</td>
<td>not specialized</td>
</tr>
</tbody>
</table>

### Patient

<table>
<thead>
<tr>
<th><strong>ParticipantObjectTypeCode</strong></th>
<th>M</th>
<th>“1” (Person)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParticipantObjectTypeCodeRole</td>
<td>M</td>
<td>“1” (Patient)</td>
</tr>
<tr>
<td>ParticipantObjectDataLifeCycle</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectIDTypeCode</td>
<td>M</td>
<td>EV(2, RFC-3881, “Patient Number”)</td>
</tr>
<tr>
<td>ParticipantObjectSensitivity</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectID</td>
<td>M</td>
<td>The patient ID in HL7 CX format.</td>
</tr>
<tr>
<td>ParticipantObjectName</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectQuery</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectDetail</td>
<td>U</td>
<td>not specialized</td>
</tr>
</tbody>
</table>

### Query Parameters

<table>
<thead>
<tr>
<th><strong>ParticipantObjectTypeCode</strong></th>
<th>M</th>
<th>“2” (system object)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParticipantObjectTypeCodeRole</td>
<td>M</td>
<td>“24” (query)</td>
</tr>
<tr>
<td>ParticipantObjectDataLifeCycle</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectIDTypeCode</td>
<td>M</td>
<td>EV(“PHARM-1”, “IHE Transactions”, “Query Pharmacy Documents”)</td>
</tr>
<tr>
<td>ParticipantObjectSensitivity</td>
<td>U</td>
<td>not specialized</td>
</tr>
<tr>
<td>ParticipantObjectID</td>
<td>M</td>
<td>Stored Query ID (UUID)</td>
</tr>
<tr>
<td>ParticipantObjectName</td>
<td>C</td>
<td>If known the value of <code>&lt;ihe:HomeCommunityId/&gt;</code></td>
</tr>
<tr>
<td>ParticipantObjectQuery</td>
<td>M</td>
<td>The AdhocQueryRequest, base64 encoded.</td>
</tr>
</tbody>
</table>

**ParticipantObjectDetail**

In one element, set “QueryEncoding” as the value of the attribute type, Set the attribute value to the character encoding, such as “UTF-8”, used to encode the ParticipantObjectQuery before base64 encoding.

In another element, set “urn:ihe:iti:xca:2010:homeCommunityId” as the value of the attribute type and the value of the homeCommunityID as the value of the attribute value, if known.
3.1.5.1.(z) Actor Specific Security Considerations

No information available yet.
4 Workflow Definitions

The management of the workflow related to clinical process has becoming a fundamental topic with the increasing of the use by different sectors of document sharing related IHE profiles with their different types of document and information.

4.1 Community Medication Prescription and Dispense Workflow Definition (CMPD-WD)

The management of the workflow related to the CMPD Profile is involved in much clinical and organizational process for its important role in the process of digitalization. The lack of a workflow management blocks the use of the Prescription in an extended way. The definition of a workflow with defined rules and tasks is needed in a scenario cross enterprise in which many actors are involved in the same process.

In this chapter a set of rules which defines the workflow of the CMPD process and the relationship with the actors involved are described. If real-world scenarios need a technical workflow management the actors involved in the process can use the “Workflow Management” Option which groups the CMPD actors with the XDW actors.

The ITI XDW Profile is a core component of a common, workflow-independent interoperability infrastructure that provides a platform upon which a wide range of specific workflows can be defined by “content specialization” with minimal specification and implementation efforts by the different domains. For the definition of the CMPD workflow it is possible to use the ITI XDW Profile as an infrastructure layer to define a set logical or clinical tasks definitions and rules to apply. The rules in the workflow definition ensure that the different participants in a workflow operate jointly to advance within tasks and to move from one task to another in a consistent way.

To integrate the CMPD Profile with ITI XDW Profile it is necessary to introduce the integrations described in the follow paragraphs.

4.1.1 Actors and Grouping

If the “Workflow Management” Option is supported the following CMPD actors shall be grouped with XDW actors to allow access and manipulation of the XDW-WD (XDW Workflow document).

<table>
<thead>
<tr>
<th>Actor</th>
<th>Groups with</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medication Treatment Planner</td>
<td>XDW: Content Creator</td>
<td>The Medication Treatment Plan Planner Actor shall create the XDW-WD to start the process. It also consumes and may be updates the XDW-WD document in case of modification to the Medication Treatment Plan.</td>
</tr>
<tr>
<td></td>
<td>XDW: Content Consumer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XDW: Content Updater</td>
<td></td>
</tr>
</tbody>
</table>

43 If “Medication Treatment Planning” Option is used.
### 4.1.2 XDW Workflow Document – Common Attributes

The CMPD Workflow Definition does not introduce new metadata and all the metadata elements used are the common XDS document metadata specified in ITI TF-3:4.1.5 and in ITI TF-3:5.4.6. In this section only the use of some specific metadata for the use of XDW in the CMPD context is specified.

<table>
<thead>
<tr>
<th>XDSDocumentEntry Attribute</th>
<th>Definition</th>
</tr>
</thead>
</table>
| typeCode                   | For the Workflow Document which tracks the CMPD process the code for the typeCode shall be:  
  **Scenario 1:** urn:ihe:pharm:cmpdwd1:2011  
  **Scenario 2:** urn:ihe:pharm:cmpdwd2:2011  
  Note: see chapter 4.1.3 for a description of the two workflow scenarios.  
  This code is the same code that shall be used in the element workflowDefinitionReference inside the Workflow Document |
| eventCodeList              | Rule 1:  
  A CMPD workflow shall be created with code OPEN and shall remain in this status until it is set to CLOSE.  
  
  Rule 2:  
  A CMPD workflow should be set to CLOSE by any actor which discovers that the partial workflow of each Medication Treatment Plan Item (for which a task “Plan” has been created) or Prescription Item within (for which a task “Ordering” has been created) has ended either by a complete dispense of the item or any other way (e.g., a cancelation, etc.). |
### XDSDocumentEntry Attribute

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceStartTime</td>
<td>It is the time at which work began on the earliest task for this workflow.</td>
</tr>
<tr>
<td>serviceStopTime</td>
<td>It is the time at which the status of the overall Workflow is changed from OPEN to CLOSE. It shall be empty when the workflow is still in OPEN state.</td>
</tr>
</tbody>
</table>

### 4.1.3 Workflow Task Definition

This chapter describes Workflow Tasks which are used in the XDW Workflow document to express a Community Pharmacy workflow.

The Community Pharmacy workflow can be divided in two different scenarios:

**Scenario 1: Including validation step**

![Figure 4.1.3-1: Scenario 1: Overall context of the workflow](image-url)

**Figure 4.1.3-1: Scenario 1: Overall context of the workflow**
Scenarios 2: Not including validation step

Please refer to Volume 1, chapter 4.4 CMPD Process Flow for a detailed explanation of the Community Pharmacy workflow scenarios. Note that in both scenarios the “Ordering” task may be triggered by an initial “Planning” task as described in section 4.4.2 of Volume 1. This task being optional and having no impact on the subsequent workflow, it is not included in the drawings above.

The following workflow tasks are defined and specified in detail in the following chapters:

- Planning
- Ordering
- Validation
- Dispensing

**Important Note:** When referencing input and output documents in tasks, both document uniqueId and homeCommunityId of the document shall be used.
4.1.3.1 Task: Planning

The task “Planning” starts the Community Pharmacy workflow by creation of the Workflow document together with a single Planning document. The task is able to record the creation of a single Medication Treatment Plan Item.

Figure 4.1.3.1-1: Planning containing Medication Treatment Plan Item leading to workflow tasks “Planning”

<table>
<thead>
<tr>
<th>Task attributes</th>
<th>Rules for the task “Planning”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task id</td>
<td>Unique id of the instance of the task</td>
</tr>
<tr>
<td>Task type</td>
<td>Planning</td>
</tr>
<tr>
<td>Task name</td>
<td>Planning_of_Medication_Treatment_Plan_Item</td>
</tr>
</tbody>
</table>
| Task description| The description element shall contain the MedicationTreatmentPlanItemId, this task is referring to (substanceAdministration/id element of the Medication Treatment Plan Item). Format compliant to the HL7 v2 CX datatype: 

*Variant 1: Only id/@root is given*

$desc = substanceAdministration/id/@root

*Variant 2: id/@root and id/@extension is given*
### Task attributes | Rules for the task “Planning”

<table>
<thead>
<tr>
<th>Task attributes</th>
<th>Rules for the task “Planning”</th>
</tr>
</thead>
<tbody>
<tr>
<td>$desc = concat(</td>
<td></td>
</tr>
<tr>
<td>substanceAdministration/id/@extension, &quot;^^^&amp;&quot;,</td>
<td></td>
</tr>
<tr>
<td>substanceAdministration/id/@root, &quot;,&amp;ISO&quot;)</td>
<td></td>
</tr>
</tbody>
</table>

### Task dependencies

**Workflow scenario 1**  
Ancestors: None  
Successors: Ordering, Dispensing

**Workflow scenario 2**  
Ancestors: None  
Successors: Ordering, Dispensing

### Status allowed

**COMPLETED**  
A Planning task is always set to COMPLETED.

### Status transactions

None

### Input

- **Optional**
  - All documents useful to understand the reason for the planning (clinical reports, ...) may be referenced.

### Output

- **Required**  
  - The Medication Treatment Plan document produced shall be referenced.

### Owner

Same Physician or organization that creates the Medication Treatment Plan document

### Owner changes

No

### Example XML for this XDW task:

```xml
<ns3:taskData>
  <ns2:taskDetails>
    <ns2:id>urn:oid:1.1.1.1.0</ns2:id>
    <ns2:taskType>Planning</ns2:taskType>
    <ns2:name>Planning_of_Medication_Treatment_Plan_Item</ns2:name>
    <ns2:status>COMPLETED</ns2:status>
    <ns2:actualOwner>Dr. Brum</ns2:actualOwner>
    <ns2:createdTime>2006-05-04T18:13:51.0Z</ns2:createdTime>
    <ns2:createdBy>Dr. Brum</ns2:createdBy>
    <ns2:lastModifiedTime>2006-05-04T18:13:51.0Z</ns2:lastModifiedTime>
    <ns2:renderingMethodExists>false</ns2:renderingMethodExists>
  </ns2:taskDetails>
</ns3:taskData>
```
4.1.3.2 Task: Ordering

The task “Ordering” starts the Community Pharmacy workflow by creation of the Workflow document together with a single Prescription document, unless already started by the optional planning task. The task is able to record the creation of a single Prescription Item (within a Prescription), hence separate tasks have to be created for each Prescription Item of the Prescription.
Figure 4.1.3.2-1: Prescription containing Prescription Items leading to workflow tasks “Ordering”

Table 4.1.3.2-1: Ordering Task Rules

<table>
<thead>
<tr>
<th>Task attributes</th>
<th>Rules for the task “Ordering”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task id</td>
<td>Unique id of the instance of the task</td>
</tr>
<tr>
<td>Task type</td>
<td>Ordering</td>
</tr>
<tr>
<td>Task name</td>
<td>Order_of_Prescription_Item</td>
</tr>
<tr>
<td>Task description</td>
<td>The description element shall contain the PrescriptionItemId, this task is referring to (substanceAdministration/id element of the Prescription Item). Format compliant to the HL7 v2 CX datatype:</td>
</tr>
<tr>
<td></td>
<td><strong>Variant 1: Only id/@root is given</strong></td>
</tr>
<tr>
<td></td>
<td>$desc = substanceAdministration/id/@root</td>
</tr>
<tr>
<td></td>
<td><strong>Variant 2: id/@root and id/@extension is given</strong></td>
</tr>
<tr>
<td></td>
<td>$desc = concat(</td>
</tr>
<tr>
<td></td>
<td>substanceAdministration/id/@extension, &quot;^^^&quot;,</td>
</tr>
<tr>
<td></td>
<td>substanceAdministration/id/@root, &quot;&amp;ISO&quot;)</td>
</tr>
<tr>
<td>Task dependencies</td>
<td><strong>Workflow scenario 1</strong></td>
</tr>
<tr>
<td></td>
<td>Ancestors: None, Planning, Validation</td>
</tr>
<tr>
<td></td>
<td>Successors: Validation</td>
</tr>
<tr>
<td></td>
<td><strong>Workflow scenario 2</strong></td>
</tr>
</tbody>
</table>
## Task attributes

<table>
<thead>
<tr>
<th><strong>Rules for the task “Ordering”</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancestors: None, Planning</td>
<td></td>
</tr>
<tr>
<td>Successors: Dispensing</td>
<td></td>
</tr>
</tbody>
</table>

### Status allowed

**COMPLETED**

An Ordering task is always set to COMPLETED.

### Status transactions

None

### input

- Optional
  - All documents useful to understand the reason for the prescription (clinical reports, …) may be referenced.

### output

- Required
  - The Prescription document produced shall be referenced.

### owner

Same Physician or organization that creates the Prescription document

### owner changes

No

### <taskEvent>

Only one

### Example XML for this XDW task:

```xml
<ns2:taskDetails>
  <ns2:id>urn:oid:1.1.1.1.1</ns2:id>
  <ns2:taskType>Ordering</ns2:taskType>
  <ns2:name>Order_of_Prescription_Item</ns2:name>
  <ns2:status>COMPLETED</ns2:status>
  <ns2:actualOwner>Dr. Brum</ns2:actualOwner>
  <ns2:createdTime>2006-05-04T18:13:51.0Z</ns2:createdTime>
  <ns2:createdBy>Dr. Brum</ns2:createdBy>
  <ns2:lastModifiedTime>2006-05-04T18:13:51.0Z</ns2:lastModifiedTime>
  <ns2:renderingMethodExists>false</ns2:renderingMethodExists>
</ns2:taskDetails>
```

```xml
<ns2:description>4711^^^&amp;1.2.3.4.5.6.7.8.9&amp;ISO</ns2:description>
```

```xml
<ns2:input>
  <ns2:part name="document">
    <!-- Document useful to understand the reason of the prescription -->
  </ns2:part>
```
<!-- uid: the document uniqueId, home: the homeCommunityId -->
<reference uid="urn:oid:1.2.3.4.4.3.2.2.3" home="urn:oid:1.2.3"/>
</ns2:part>
</ns2:input>

<!-- output documents -->
<ns2:output>
  <ns2:part name="Prescription_Document">
    <!-- Prescription document according to PRE Profile -->
    <!-- uid: the document uniqueId, home: the homeCommunityId -->
    <reference uid="urn:oid:1.2.3.4.4.4" home="urn:oid:1.2.3"/>
  </ns2:part>
</ns2:output>
4.1.3.3 Task: Validation

The task “Dispensing” is able to record the dispense of a single Prescription Item (within a Prescription) in an existing workflow document. Each dispense results in the creation of a Medication Dispense document. Hence separate tasks have to be created for each Prescription Item of the Prescription which is dispensed.

In case of dispensing a single medication which was not prescribed (e.g., OTC medication), the dispensing act starts the workflow. Hence the task is recorded in a newly created workflow document.

Figure 4.1.3.3-1: Prescription Items leading to workflow tasks “Validation”

<table>
<thead>
<tr>
<th>Task attributes</th>
<th>Rules for the task “Validation”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task id</td>
<td>Unique id of the instance of the task</td>
</tr>
<tr>
<td>Task type</td>
<td>Validation</td>
</tr>
<tr>
<td>Task name</td>
<td>Validation_of_Prescription_Item</td>
</tr>
<tr>
<td>Task description</td>
<td>The description element shall contain the PrescriptionItemId, this task is referring to (substanceAdministration/id element of the Prescription Item). Format compliant to the HL7 v2 CX datatype:</td>
</tr>
</tbody>
</table>
### Task attributes

<table>
<thead>
<tr>
<th>Rules for the task “Validation”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variant 1:</strong> Only id/@root is given</td>
</tr>
<tr>
<td>$desc = substanceAdministration/id/@root</td>
</tr>
<tr>
<td><strong>Variant 2:</strong> id/@root and id/@extension is given</td>
</tr>
<tr>
<td>$desc = concat(phantomvalue, substanceAdministration/id/@extension, &quot;^^^&amp;&quot;, substanceAdministration/id/@root, &quot;&amp;ISO&quot;)</td>
</tr>
</tbody>
</table>

### Task dependencies

<table>
<thead>
<tr>
<th>Workflow scenario 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancestors: Ordering</td>
</tr>
<tr>
<td>Successors: Dispensing, Ordering</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workflow scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not applicable</td>
</tr>
</tbody>
</table>

### Status allowed

<table>
<thead>
<tr>
<th>Status allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IN PROGRESS</strong></td>
</tr>
<tr>
<td>A Validation task shall be set to IN PROGRESS, when the outcome of the task is a preliminary validation result. A Pharmaceutical Advice document is preliminary, if the element statusCode is set to active.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMPLETED</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Validation task shall be set to COMPLETED, when the outcome of the task is a final validation result. A Pharmaceutical Advice document is final, if the element statusCode is set to completed.</td>
</tr>
</tbody>
</table>

Note: See Pharmacy Pharmaceutical Advice (PADV) Profile, Vol. 2, chapter “6.3.4.3.6 Status Code”.

### Status transactions

<table>
<thead>
<tr>
<th>Status transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>All the states of the task (in progress, completed) are performed in sequence</td>
</tr>
</tbody>
</table>

### input

- **Required**
  - The Prescription document containing the validated Prescription Item shall be referenced. The specific Prescription Item the task is referring to shall also be stated in that structure.
  - The ancestor task shall be referenced.

- **Optional**
  - All additional documents useful to understand the reason for the outcome of the Pharmaceutical Advice may be referenced.

### output

- **Required**
  - The Pharmaceutical Advice document produced shall be referenced.

### owner

- Same Physician or organization that creates the Pharmaceutical Advice document
Task attributes | Rules for the task “Validation”
--- | ---
changes of task owner | Yes
The owner may change, if more than one Pharmaceutical Advice documents are created (in case of a multi-step validation using preliminary validation).

Example XML for this XDW task:

```xml
<ns3:taskData>
  <ns2:taskDetails>
    <ns2:id>urn:oid:2.2.2.2.2</ns2:id>
    <ns2:taskType>Validation</ns2:taskType>
    <ns2:name>Validation of Prescription Item</ns2:name>
    <ns2:status>COMPLETED</ns2:status>
    <ns2:actualOwner>Dr. Brum</ns2:actualOwner>
    <ns2:createdTime>2006-05-04T18:13:51.0Z</ns2:createdTime>
    <ns2:createdBy>Dr. Brum</ns2:createdBy>
    <ns2:lastModifiedTime>2006-05-04T18:13:51.0Z</ns2:lastModifiedTime>
    <ns2:renderingMethodExists>false</ns2:renderingMethodExists>
  </ns2:taskDetails>
  <!-- The description element shall contain the PrescriptionItemId, this task is referring to (substanceAdministration/id element of the Prescription Item) -->
  <ns2:description>4711^^^&amp;1.2.3.4.5.6.7.8.9&amp;ISO</ns2:description>
  <!-- input documents -->
  <ns2:input>
    <ns2:part name="Prescription_Document">
      <!-- Prescription document according to PRE Profile -->
      <!-- uid: the document uniqueId, home: the homeCommunityId -->
      <reference uid="urn:oid:1.2.3.4.4.4" home="urn:oid:1.2.3"/>
    </ns2:part>
    <ns2:part name="Ancestor_task">
      <!-- Ancestor task -->
      <reference taskId="urn:oid:1.1.1.1"/>
    </ns2:part>
  </ns2:input>
  <!-- output documents -->
  <ns2:output>
    <ns2:part name="Pharmaceutical_Advice_Document">
      <!-- Pharmaceutical Advice document according to PADV Profile -->
    </ns2:part>
  </ns2:output>
</ns3:taskData>
```
--- uid: the document **uniqueId**, home: the **homeCommunityId** ---

<reference uid="urn:oid:1.2.3.4.4.5" home="urn:oid:1.2.3"/>

</ns2:part>

</ns2:output>

</ns3:taskData>
4.1.3.4 Task: Dispensing

The task “Dispensing” is able to record the dispense of a single Prescription Item (within a Prescription). Each dispense results in the creation of a Medication Dispense document. Hence separate tasks have to be created for each Prescription Item of the Prescription which is dispensed.

![Diagram showing Dispensing process]

**Figure 4.1.3.4-1: Prescription Items leading to workflow tasks “Dispensing”**

<table>
<thead>
<tr>
<th>Task attributes</th>
<th>Rules for the task “Dispensing”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task id</td>
<td>Unique id of the instance of the task</td>
</tr>
<tr>
<td>Task type</td>
<td>Dispensing</td>
</tr>
<tr>
<td>Task name</td>
<td>Dispense_of_Prescription_Item</td>
</tr>
<tr>
<td>Task description</td>
<td>The description element shall contain the PrescriptionItem.id, this task is referring to (substanceAdministration/id element of the Prescription Item). Format compliant to the HL7 v2 CX datatype: Variant.1::<a href="mailto:Only@root.is.given">Only@root.is.given</a></td>
</tr>
</tbody>
</table>
### Task attributes

<table>
<thead>
<tr>
<th>Rules for the task “Dispensing”</th>
</tr>
</thead>
<tbody>
<tr>
<td>$desc = substanceAdministration/id/@root</td>
</tr>
<tr>
<td><strong>Variant 2:</strong> id/@root and id/@extension is given</td>
</tr>
<tr>
<td>$desc = concat(</td>
</tr>
<tr>
<td>substanceAdministration/id/@extension, &quot;^^^&amp;&quot;,</td>
</tr>
<tr>
<td>substanceAdministration/id/@root, &quot;&amp;ISO&quot;)</td>
</tr>
</tbody>
</table>

### Task dependencies

<table>
<thead>
<tr>
<th>Workflow scenario 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancestors: None, Planning, Validation</td>
</tr>
<tr>
<td>Successors: None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workflow scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancestors: None, Planning, Ordering</td>
</tr>
<tr>
<td>Successors: None</td>
</tr>
</tbody>
</table>

### Status allowed

<table>
<thead>
<tr>
<th>IN PROGRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Dispensing task shall be set to IN PROGRESS, when the order is repeatable and the current dispense is not the last repeat.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMPLETED</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Dispensing task shall be set to IN PROGRESS, when the order is repeatable and the current dispense is the last repeat.</td>
</tr>
</tbody>
</table>

Note: See Pharmacy Prescription (PRE) Profile, Vol. 2, chapter “6.3.4.1.3.21 Amount of units of the consumable to dispense”.

### Status transactions

All the states of the task (in progress, completed) are performed in sequence.

### input

- Required
  - The Prescription document containing the dispensed Prescription Item shall be referenced. The specific Prescription Item the task is referring to shall also be stated in that structure.
  - Additional in scenario 2: The Pharmaceutical Advice document related to the Prescription Item which approved the dispensing act shall be referenced.
  - The ancestor task shall be referenced.

### output

- Required
  - The Medication Dispense document produced shall be referenced in the Output element of this task.

### owner

Same Physician or organization that creates the Dispense document.

### changes of task owner

Yes
The owner may change, if more than one Dispense documents are created (in case of a repeatable order).

### <taskEvent>

At least one
Example XML for this XDW task:

```
<ns3:taskData>
  <ns2:taskDetails>
    <ns2:id>urn:oid:3.3.3.3</ns2:id>
    <ns2:taskType>Dispensing</ns2:taskType>
    <ns2:name>Dispense_of_Prescription_Item</ns2:name>
    <ns2:status>COMPLETED</ns2:status>
    <ns2:actualOwner>Dr. Brum</ns2:actualOwner>
    <ns2:createdTime>2006-05-04T18:13:51.0Z</ns2:createdTime>
    <ns2:createdBy>Dr. Brum</ns2:createdBy>
    <ns2:lastModifiedTime>2006-05-04T18:13:51.0Z</ns2:lastModifiedTime>
    <ns2:renderingMethodExists>false</ns2:renderingMethodExists>
  </ns2:taskDetails>
  <!-- The description element shall contain the PrescriptionItemId, this task is referring to (substanceAdministration/id element of the Prescription Item) -->
  <ns2:description>4711^^^&amp;1.2.3.4.5.6.7.8.9&amp;ISO</ns2:description>
  <!-- input documents -->
  <ns2:input>
    <ns2:part name="Prescription_Document">
      <!-- Prescription document according to PRE Profile -->
      <!-- uid: the document uniqueId, home: the homeCommunityId -->
      <reference uid="urn:oid:1.2.3.4.4.4" home="urn:oid:1.2.3"/>
    </ns2:part>
    <ns2:part name="Pharmaceutical_Advice_Document">
      <!-- Pharmaceutical Advice document according to PADV Profile -->
      <!-- uid: the document uniqueId, home: the homeCommunityId -->
      <reference uid="urn:oid:1.2.3.4.4.5" home="urn:oid:1.2.3"/>
    </ns2:part>
    <ns2:part name="Ancestor_task">
      <!-- Ancestor task -->
      <reference taskId="urn:oid:2.2.2.2"/>
    </ns2:part>
  </ns2:input>
  <!-- output documents -->
  <ns2:output>
    <ns2:part name="Dispense_Document">
      <!-- Dispense document according to DIS Profile -->
      <!-- uid: the document uniqueId, home: the homeCommunityId -->
      <reference uid="urn:oid:1.2.3.4.4.6" home="urn:oid:1.2.3"/>
    </ns2:part>
  </ns2:output>
</ns3:taskData>
```
Glossary

Add the following terms to the Glossary:

No new terms