

KEREVAL

4 rue Hélène Boucher
Z.A Bellevue
35 235 THORIGNE FOUILLARD - FRANCE

Tél. : +33 (0) 223 203 664

RCS : B 442 789 210

APE : 722 C



KEREVAL HEALTHLAB - Project IHE EUROPE

User Guide

Gazelle Master Model – V4.x

Version : 1.02

Date: 04/09/2014

Author: Thomas DOLOUE

Function: Quality Assistant

Reference:

KER3-MAN-HEALTHLAB-
GAZELLE_MASTER_MODEL_USER-1.02

Status: approved



■ **KEREVAL Approval**

Name	Function	Date	Visa
Eric POISEAU	Lab Manager	04/09/2014	OK

■ **Diffusion**

Internal	Recipient	Date	Exemplary
KEREVAL	HealthLab	04/09/2014	Electronic version

External	Recipient	Date	Exemplary
GMM users		04/09/2014	Electronic version

■ **Document history**

Version	Date	Author	Modifications
V0.01	02/09/2014	Thomas DOLOUE	Creation
V1.01	04/09/2014	Thomas DOLOUE	For review
V1.02	04/09/2014	Eric POISEAU	Approve

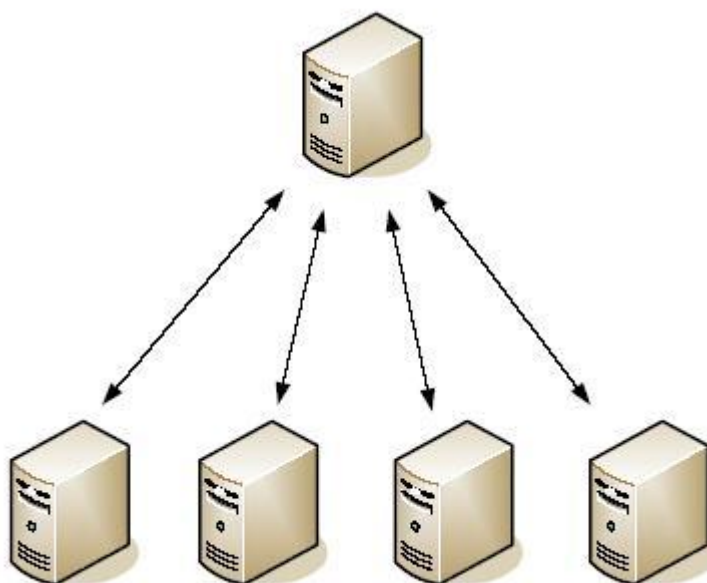
■ **Table of content**

1	INTRODUCTION	4
1.1	Gazelle Master Model	4
1.2	Edition of the Technical Framework Concepts / Test definitions / Meta Tests / Dependencies. (available)	4
1.3	Sharing of Technical Framework (available)	4
1.4	Sharing of Test definitions (available)	4
1.5	Sharing of samples (available)	4
2	TF – OVERVIEW	6
2.1	Project overview	6
2.2	Web user interface	6
2.2.1	Description	6
2.2.2	Navigation	7
3	TF - INTEGRATION PROFILE DIAGRAM	8
3.1	Project overview	8
3.2	Technical Framework Web user interface	8
3.2.1	Description	8
4	EDITING PROFILE INFORMATION	9
4.1	Introduction	9
4.2	Add new actors	9
4.3	Add new transaction	11
4.4	Link Transactions to Actors	14
5	TEST DEFINITION	19
5.1	1. Test Summary	20
5.2	Test Description	21
5.3	Test Roles	21
5.4	Test Steps	23
6	META TESTS	27
7	CONFIGURATION OF MASTER MODEL SLAVES	28
7.1	Pre-requisite	28
7.2	Initialisation of slony on the master system	28
7.3	Initialization of slony on the slaves	36
7.4	Launching the synchronization or restarting the synchronization	37

1 Introduction

The gazelle master model manages the sharing of the model information to be used by the different Gazelle instances. Gazelle database consist of more than 190 tables. Gazelle instances are running as slaves of the master model and can request updates from the master.

1.1 Gazelle Master Model



Instances of Gazelle

1.2 Edition of the Technical Framework Concepts / Test definitions / Meta Tests / Dependencies. (available)

Module that allows the user to create/read/update/delete/deprecate concepts in the master data model.

1.3 Sharing of Technical Framework (available)

Each gazelle instance can get the update of the Technical Framework concepts from the master models.

1.4 Sharing of Test definitions (available)

As for IHE Technical Framework concepts, sharing of test definitions is possible through the Gazelle Master Model.

1.5 Sharing of samples (available)

Samples are used by the connect-a-thon participants to share images and documents between the creator and reader without using transactions. Files are stored into Gazelle and can be downloaded by other users. Numerous types of samples are defined, the ones are stored in Gazelle Master Model.

Sharing of Links to technical referencies (available) Link (URL) to reference documents can be associated to Domain, Profile, Transactions and tuple Actor/Profile/Option. Those links are share through GMM with the clients.

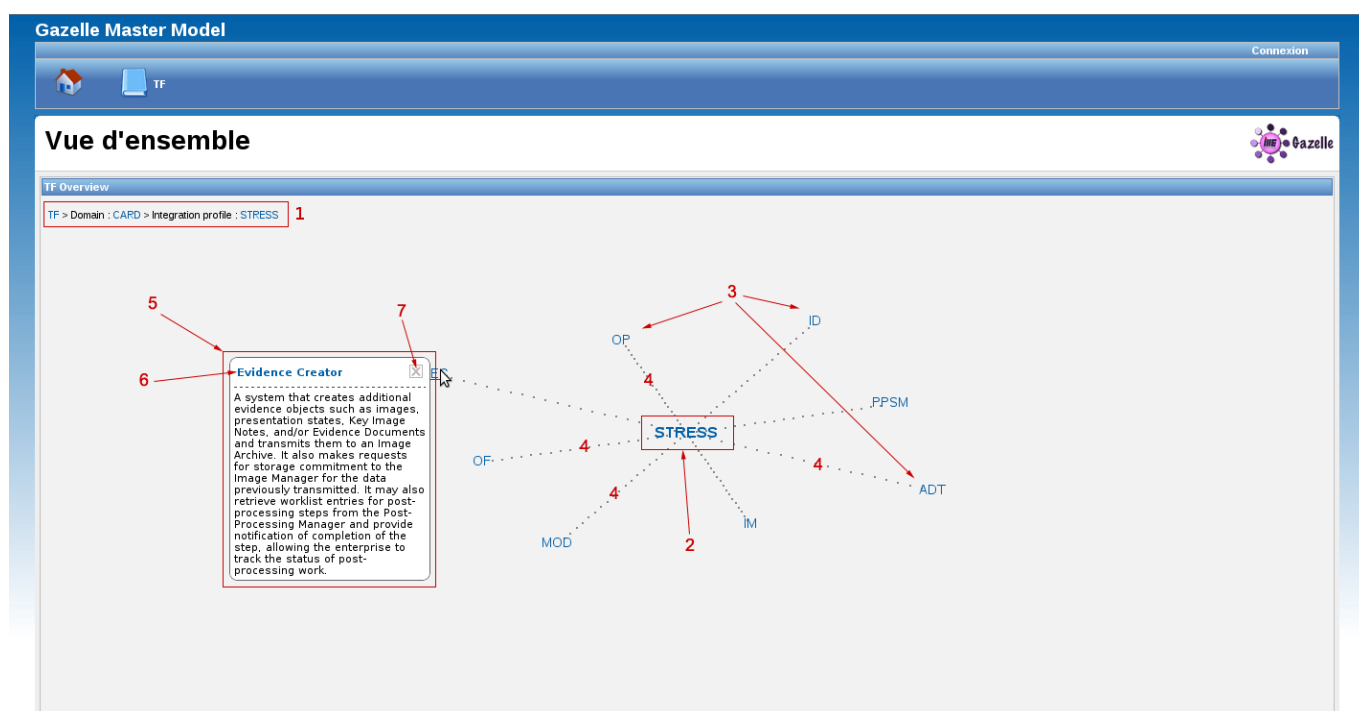
2 TF – Overview

2.1 Project overview

The Technical Framework (TF) overview is a tool that displays a graphical interface for the navigation among the TF concepts, indicating the description of those concepts and the access to their informations page.

2.2 Web user interface

2.2.1 Description



1. Breadcrumb : indicates the path in the navigation among TF concepts
2. Root : the keyword of the concept selected
3. Children : results concerning the root
4. Edge : link between the root and its children
5. Description : information about the child whose the mouse is over it
6. Link to access to the information page of the concept in the description
7. To close the description

2.2.2 Navigation

The first graphical representation displays all domains of the Technical Framework. Then, the navigation must be done in the following order :

- all integration profiles for given domain
- all actors for given integration profile
- all transactions and all integration profile options for given actor and integration profile
- all initiations and responders for given transaction

A click on the keyword of a children allows to generate the graphic. A click on the root allows to go back in the navigation and it allows to generate the previous graphic.

3 TF - Integration profile diagram

3.1 Project overview

On the information page of an integration profile, the integration profile diagram is a graphical representation that displays the transactions between the actors for this integration profile.

3.2 Technical Framework Web user interface

3.2.1 Description

Integration Profile: Eye Care Charge Posting

Informations sur le Profil d'Intégration

Id	68
Mot-clé	EYE-CHG
Nom	Eye Care Charge Posting
Description	The Charge Posting Integration Profile describes mechanisms to transfer charge information to the clinic or enterprise Charge Processing System. This section corresponds to Transaction RAD-35 of the IHE Technical Framework. Transaction RAD-35 is used by the Department System Scheduler/Order Filler and Charge Processor actors. This section also refers to RAD-36 of the IHE technical framework that is used by the ADT Patient Registration and Charge Processor Actors. This section will correlate the activities that occur in the eye care clinic with these IHE defined transactions. The standards used are Health Level Seven, Version 2.3.1: Chapter 6 - Financial Management DICOM 2004 PS 3.4 Modality Performed Procedure Step SOP Class.
Statut	Final Text

Acteurs | Couples Acteur/Transaction pour ce Profil d'Intégration | Acteurs/Options

Id	Mot-clé	Nom	Description	Action
12	ADT	ADT Patient Registration	A system responsible for adding and/or updating patient demographic and encounter information. In particular, it registers a new patient with the Order Placer and Department System.	
2	CHARGE_PRO	Charge Processor	Receives the posted charges and serves component of the financial system. Further definition of this actor is beyond current IHE scope.	
110	LEGACY_MON	Legacy Acquisition	Legacy Acquisition Modality Importer	

1. diagram of actor/transaction pairs in the selected integration profile
2. zoom out
3. zoom in
4. save the diagram on the user computer

4 Editing Profile Information

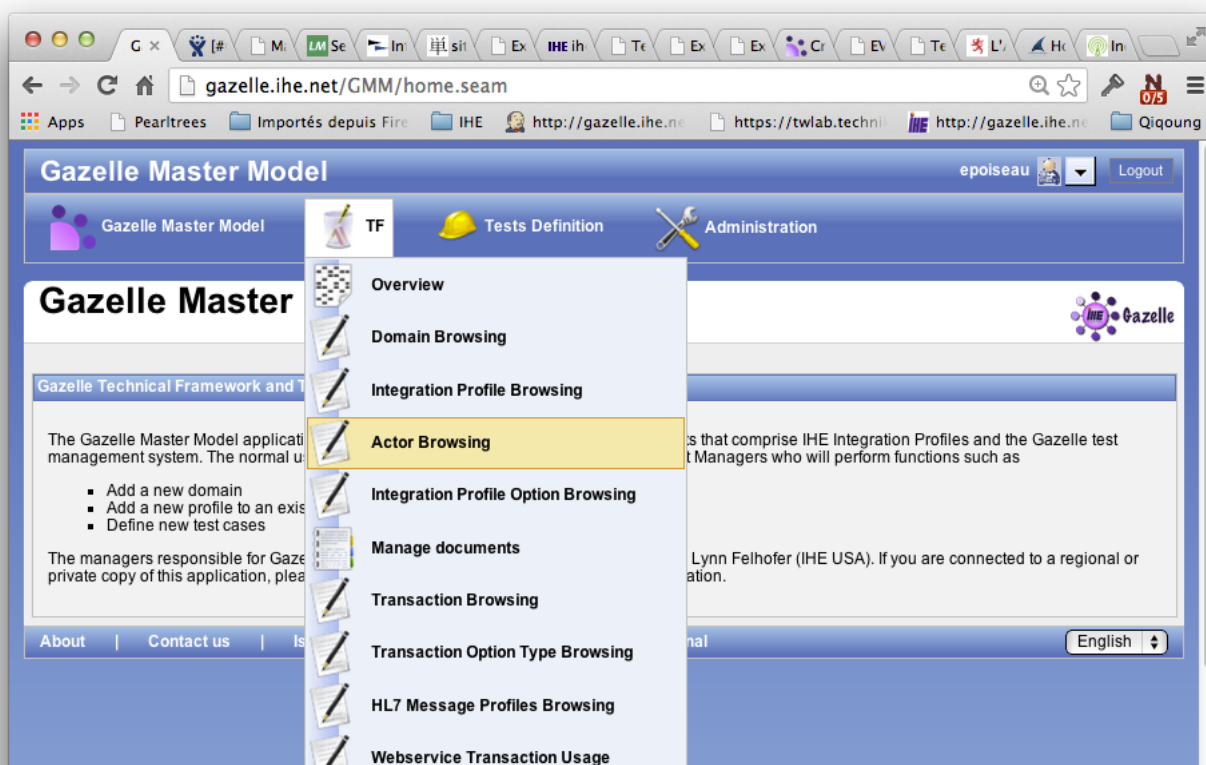
4.1 Introduction

Gazelle Master Model (GMM) allows administrators to add new Integration Profile information into Gazelle. This enables vendors to register testing these profiles at a Connectathon. Gazelle must be configured to know about the actors, transactions, and options defined within an Integration Profile. It must know which Domain the Integration Profile belongs to.

4.2 Add new actors

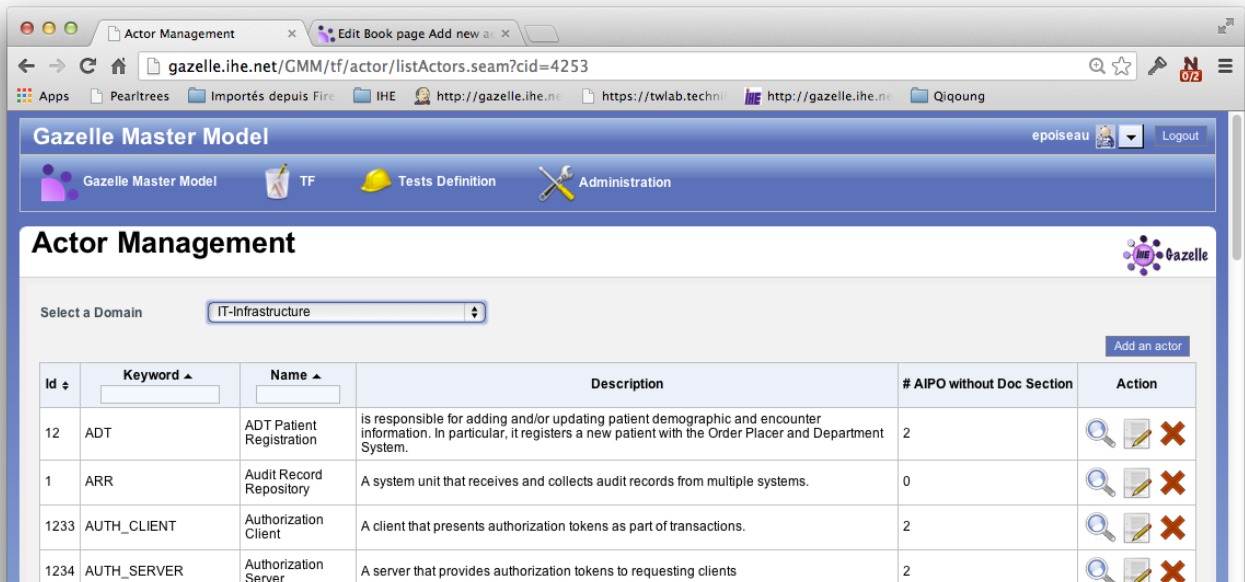
Gazelle is populated with **Actors** from Integration Profiles across all IHE domains. Prior to adding a new Actor, search the list of existing Actors to see if it already exists (eg. an Acquisition Modality or a Content Creator actor is used by many IHE Integration Profiles).

1. From the main menu, select **TF -> Actor Management**

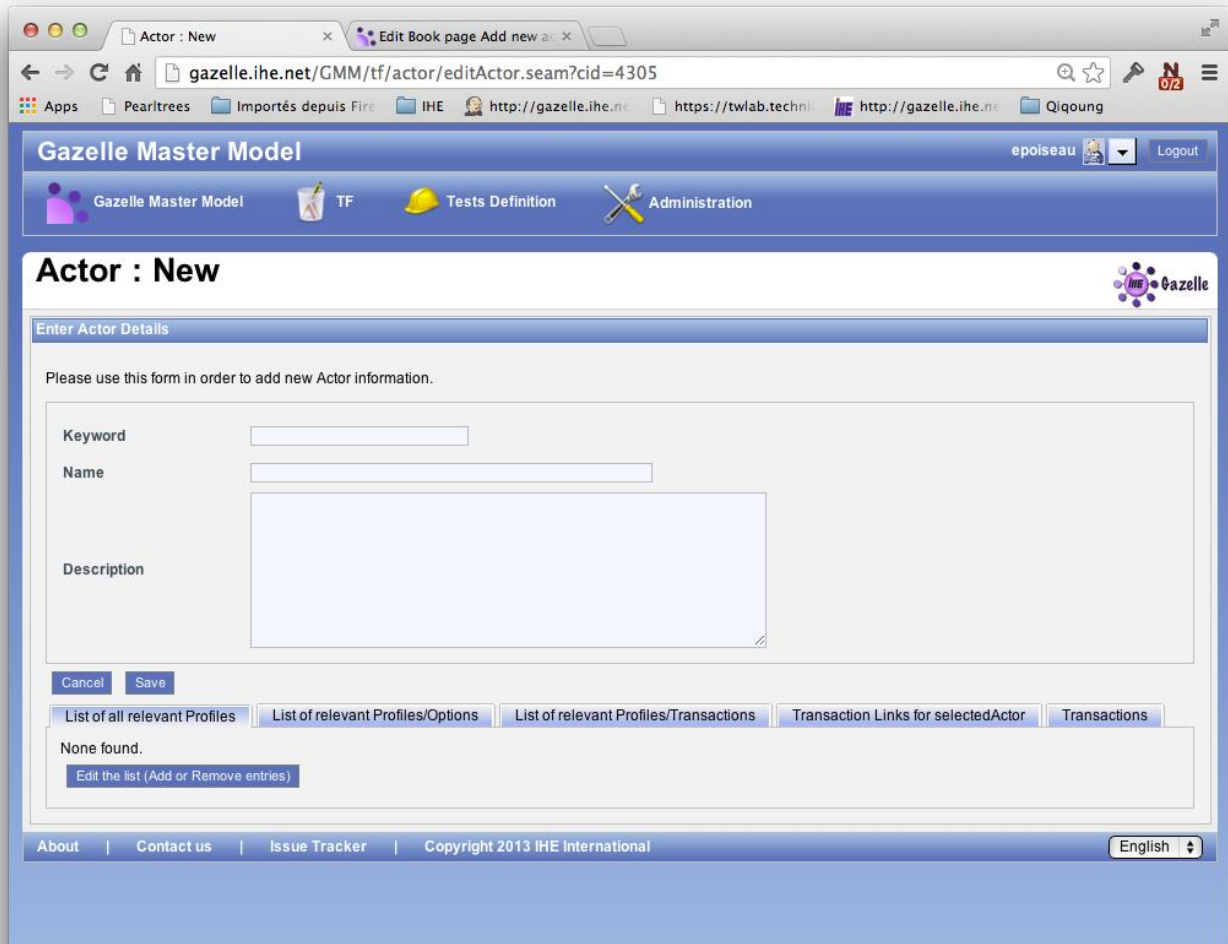


2. Search for existing Actors. There are several methods you can use:
 - a. You can restrict the search to a specific domain by selecting a value in the **Select a Domain** list box
 - b. Search by Keyword by typing into the data entry box at the top of the **Keyword** or **Name** column. (The search starts as you begin typing; there is no need to press **Enter** to start the search.)

- c. You can sort the **Keyword** or **Name** columns by clicking on the up and down triangles in the column heading.



3. To add a new actor, select the **Add an actor** button at the top right of the page.
4. On the **Actor : New** screen, enter:
- a. **Keyword** - This is a short form of the Actor name; it can be an abbreviation or acronym. Use all upper case letters and underscores. No spaces. (Although gazelle allows you to edit this keyword, once you complete all of your data entry, you should not change this keyword later. The dependency configuration relies on this value to remain the same.)
 - b. **Name** – This is the full name of the actor from the Technical Framework.
 - c. **Description** – You can copy the definition for the actor from the Technical Framework. May be left blank.

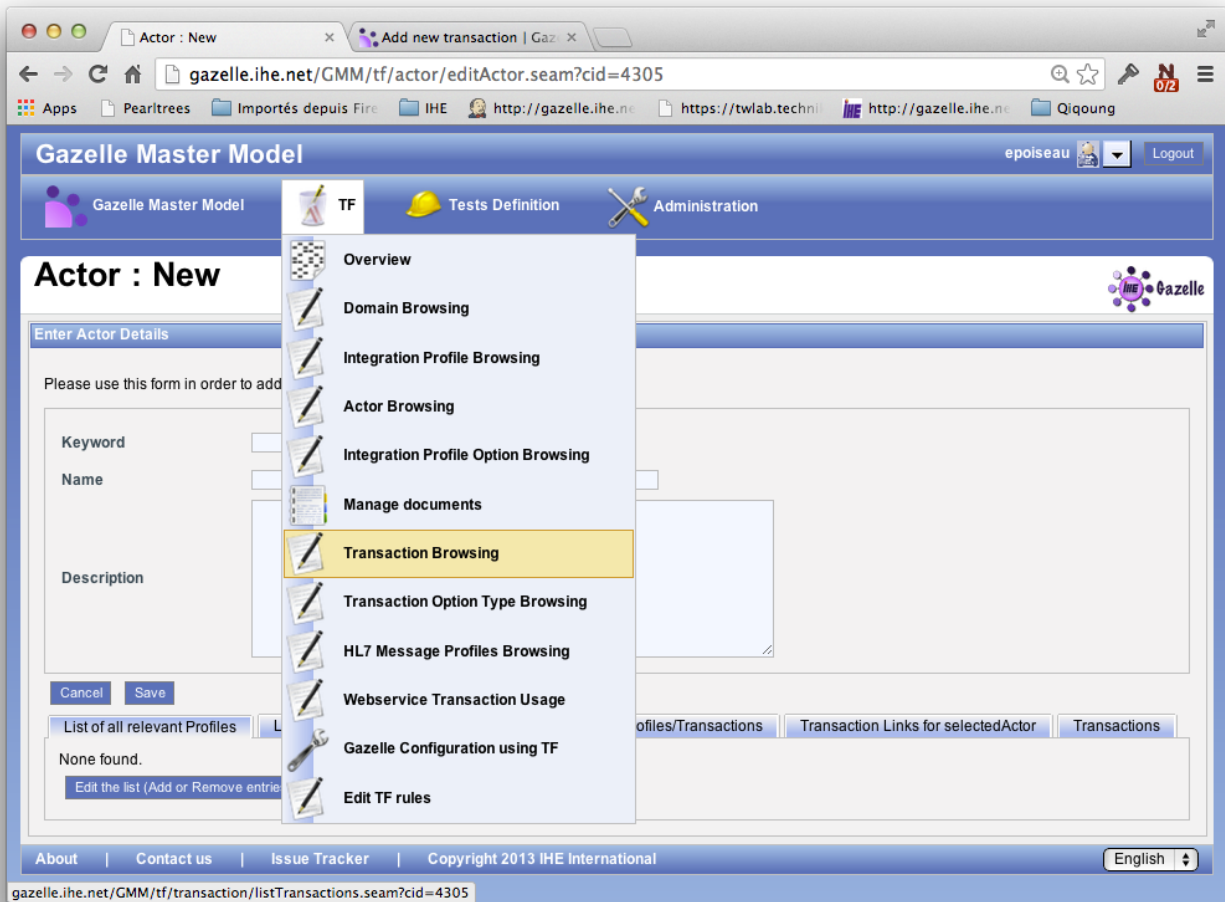


5. Select the **Save** button to finish.

4.3 Add new transaction

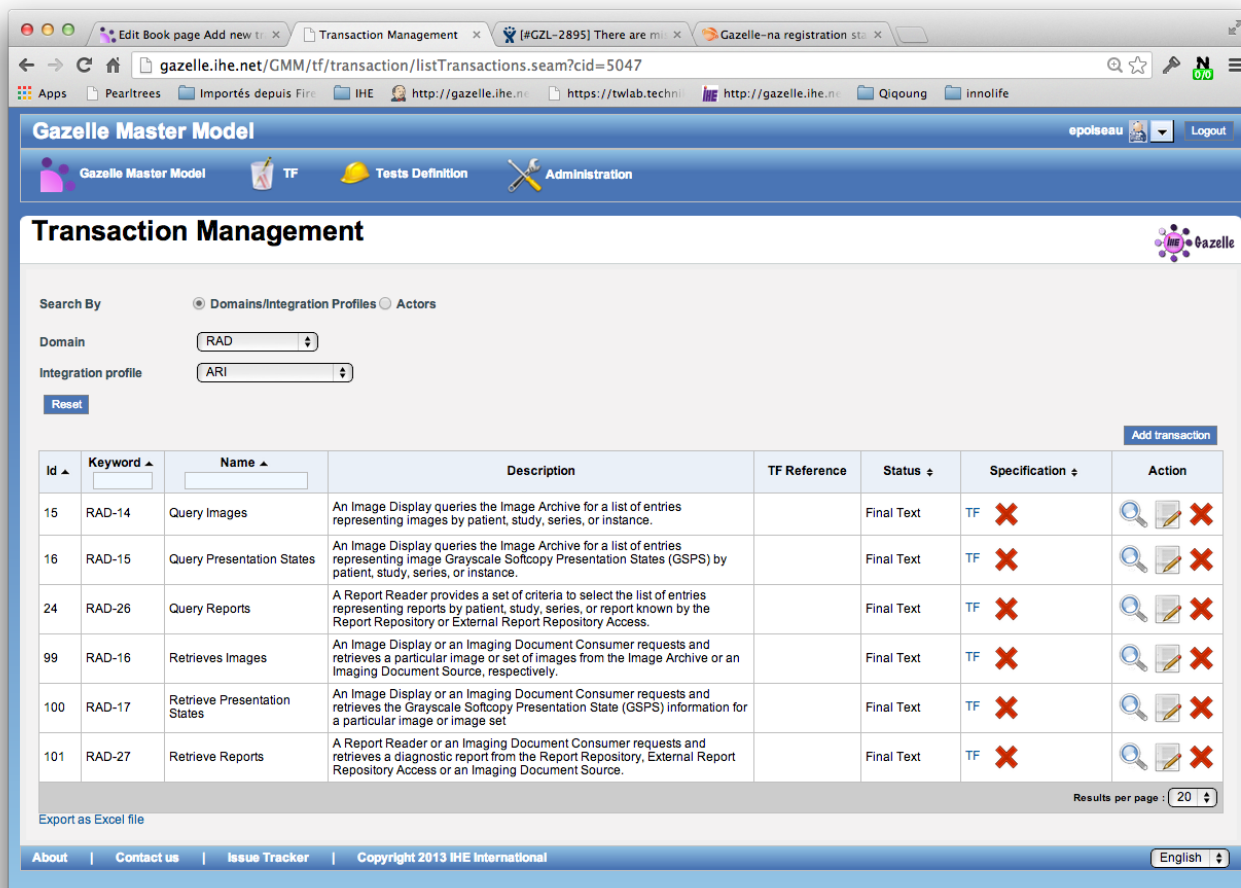
Transactions, like actors, can be viewed in a list fashion accessed from the **TF** drop down menu.

1. From the main menu, select **TF à Transaction Browsing**



2. Search for existing **Transactions**.

- a. Search by entering a **Keyword**. This is the abbreviation of the transaction, eg ITI-2, QRPH-1, RAD-8.
- b. Sort the list by **Keyword** or **Name** and page through the list using the page numbers at the bottom of the screen.



3. To add a new transaction, select the **Add transaction** button at the top right of the page.
4. On the following page, enter:
 - a. **Keyword** – This is a shortened form of the Transaction containing the domain acronym followed by a dash, then a number (no spaces) **Example:** RAD-4
 - b. **Name** – This is the full name of the transaction from the Technical Framework.
 - c. **Description** – You can copy the definition for the transaction from the Technical Framework. May be left blank.
 - d. **TF Reference** – The reference to the particular volume and section from the Technical Framework. May be left blank.
 - e. **Status** – All new transactions entered into Gazelle will be entered as **Trial Implementations**. May also be **Final Text**.

The screenshot shows the 'editTransaction.seam?cid=5001' page in the Gazelle Master Model. The page features a blue header with the title 'Gazelle Master Model' and a user profile 'epoiseau' with a 'Logout' button. Below the header is a navigation bar with icons for 'Gazelle Master Model', 'TF', 'Tests Definition', and 'Administration'. The main content area contains a form with fields for 'Keyword', 'Name', 'Description', 'TF Reference', and 'Status'. The 'Status' field is a dropdown menu currently showing 'Please Select...'. Below the form are 'Cancel', 'Save', and 'View' buttons. At the bottom of the form is a 'Section' header and a 'Document Section' field with the value 'None' and a green plus sign. The footer contains links for 'About', 'Contact us', 'Issue Tracker', and 'Copyright 2013 IHE International', along with a language selector set to 'English'.

5. Click on the **Save** button to save your changes

4.4 Link Transactions to Actors

Transactions occur between actors; one actor is the source and another is the destination. Gazelle is configured to know that a transaction is **From** an actor **To** another actor. can be viewed in a list fashion accessed from the **TF** drop down menu.

1. From the main menu, select TF -> Transaction Management

Gazelle Master Model

Transaction Management

Search By: Domains/Integration Profiles Actors

Domain:

Integration profile:

[Reset](#) [Add transaction](#)

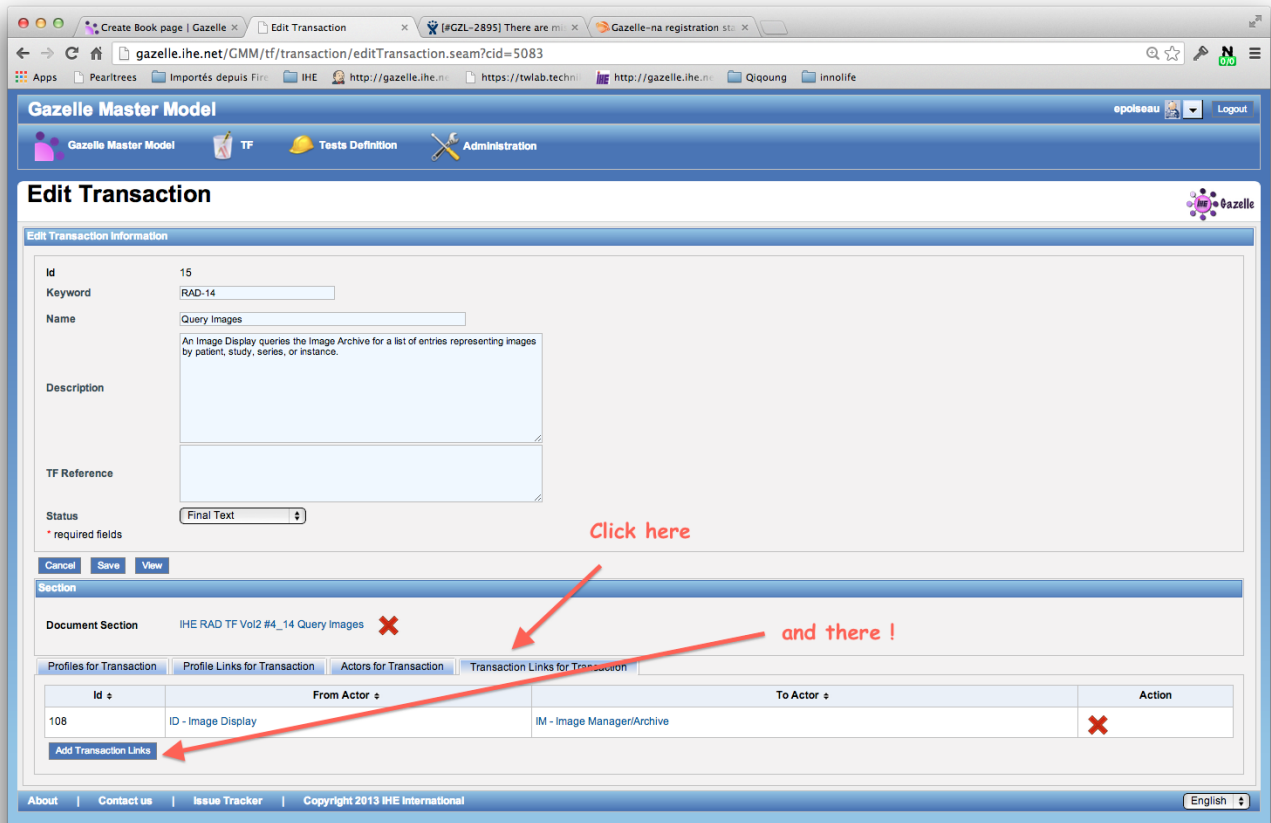
Id	Keyword	Name	Description	TF Reference	Status	Specification	Action
15	RAD-14	Query Images	An Image Display queries the Image Archive for a list of entries representing images by patient, study, series, or instance.		Final Text	TF	
16	RAD-15	Query Presentation States	An Image Display queries the Image Archive for a list of entries representing image Grayscale Softcopy Presentation States (GSPS) by patient, study, series, or instance.		Final Text	TF	
24	RAD-26	Query Reports	A Report Reader provides a set of criteria to select the list of entries representing reports by patient, study, series, or report known by the Report Repository or External Report Repository Access.		Final Text	TF	
99	RAD-16	Retrieves Images	An Image Display or an Imaging Document Consumer requests and retrieves a particular image or set of images from the Image Archive or an Imaging Document Source, respectively.		Final Text	TF	
100	RAD-17	Retrieve Presentation States	An Image Display or an Imaging Document Consumer requests and retrieves the Grayscale Softcopy Presentation State (GSPS) information for a particular image or image set.		Final Text	TF	
101	RAD-27	Retrieve Reports	A Report Reader or an Imaging Document Consumer requests and retrieves a diagnostic report from the Report Repository, External Report Repository Access or an Imaging Document Source.		Final Text	TF	

Export as Excel file

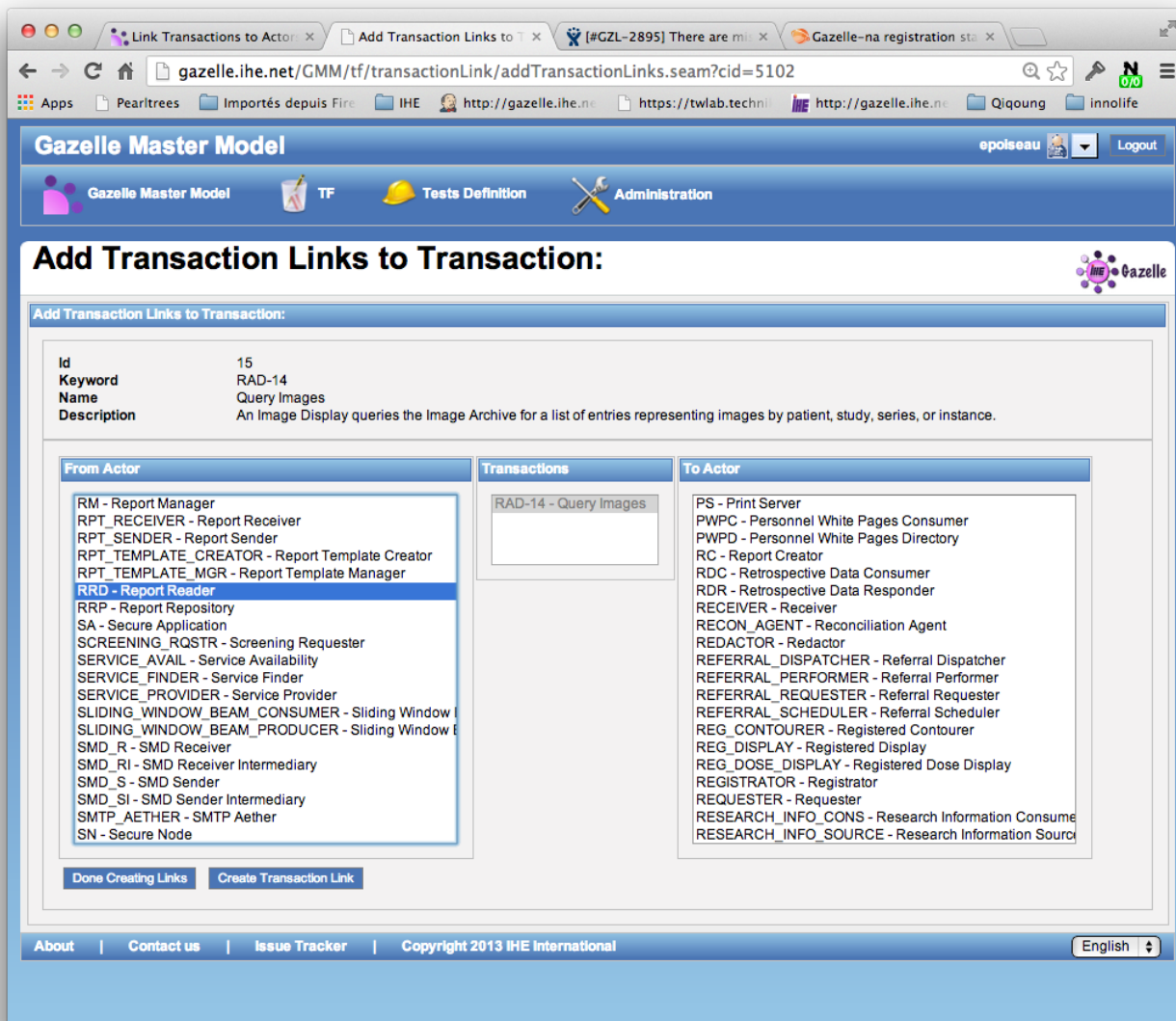
Results per page:

Copyright 2013 IHE International

2. Search for existing **Transactions**. Search by entering a **Keyword**. This is the abbreviation of the transaction, eg ITI-2, QRPH-1, RAD-8.
3. Click on the **Edit** icon in the **Action** column for the transaction you found.
4. On the **Edit Transaction** page, select the **Transactions Links for Transaction** tab, then click the **Add Transaction Links** button.



5. Select the appropriate actors from the **From Actor** and **To Actor** list.



6. Click the **Create Transaction Link** button. Repeat as needed, then click the **Done creating links** button
7. Verify the accuracy of your data entry by going back to review the **Transaction links for Transaction** tab.

Not currently covered in this document, but needed in order for profile entry to be complete:

1. Entering profile mandatory groupings (aka profile dependencies) ie Actor A in Profile P must also implement Actor B in profile Q (eg all actors in the XDS.b profile must also implement ATNA Secure Node)
2. Entering default configurations for transactions in the profile
3. Entering sample definitions

4. Entering Roles for test definitions

5 Test Definition

Test definitions are available

- in Gazelle Master Model under Tests Definition --> Tests Definition (read/write)
- in Test Management under Tests Definiton --> Tests Definition (read only)

Test definitions are, with the technical framework, the basis of Gazelle and its important feature to prepare for and participate in a connect-a-thon. The tests define the scenarios the different actors implemented by a system must pass to be validated by the connect-a-thon managers. This section of the documentation is mostly dedicated to test editors to explain the different sections of a test and how they have to be filled when creating new tests.

Prior to writing a test, three main concepts have to be introduced that determine who will see the test and when.

- **Test type:** indicates whether the test has to be performed during the connectathon or the pre-connect-a-thon period. A third type was added in order to gather the tests dedicated to HITSP profiles.
- **Test status:** indicates the level of advancement of the test. Only "ready" tests will be displayed in the participants' system's tests list during the connect-a-thon or pre-connect-a-thon period. "To be completed" are test currently under development, but are not yet ready to expose to participants. "Deprecated" tests are those which are not used anymore, in the same way the storage/substitute tests have been replaced by more relevant ones. Finally, the "convert later" status has been created when the test definitions have been imported from the Kudu system; it means that we do not need this test by now and consequently it is not a priority to work on it.
- **Test Peer Type:** indicates if the system can perform this test with "no peer" (no transaction to exchange with a vendor test partner), "peer to peer" are tests covering a subset of a profile, typically with 2, or sometimes 3, partners. "Group tests" cover a workflow within an integration profile and are tests run by a full set of actors within the profile; group test are typically supervised directly by a connectathon monitor.

Each test definitions is built of four parts which are defined below. Each part is contained on an separate tab within the test

5.1 1. Test Summary

The screenshot shows the 'Edit Test' page in a web browser. The browser address bar shows the URL: gazelle.ihe.net/GMM/testing/testsDefinition/editTestPage.seam?id=11426&cid=3473. The page title is 'Edit Test'. Below the title, there is a breadcrumb: 'Update Test (ACM_Multiple_AMs_and_ACs : ACM_Multiple_AMs_and_ACs)'. There are tabs for 'Test Summary', 'Test Description', 'Test Roles', and 'Test Steps', with 'Test Summary' selected. The form contains the following fields:

- Author of the test is:
- Keyword*:
- Name*:
- Short Description*:
- Type*:
- Status:
- Peer Type*:
- Is Orchestrable:
- Is Validated?:
- Test permanent link: <http://gazelle.ihe.net/GMM/test.seam?id=11426>
- Version:

At the bottom right of the form is a button labeled 'Update this test'. Below the form is a table titled 'User Comments' with the following data:

User	Date of creation	Last modifier	Last changed	Comment	Action
Manny	8/13/13 5:36:00 PM	Manny	8/13/13 5:36:00 PM	This is the last version prior to ACM as the Alarm Communication Management profile; it becomes the Alert Communication Management profile in 2013.	
psherman	9/19/13 8:10:24 PM	psherman	9/19/13 8:10:24 PM	This is now the first version as Alert Communication Management for 2014	

At the bottom right of the table is a 'Results per page' dropdown set to 20.

It gives general informations about the test:

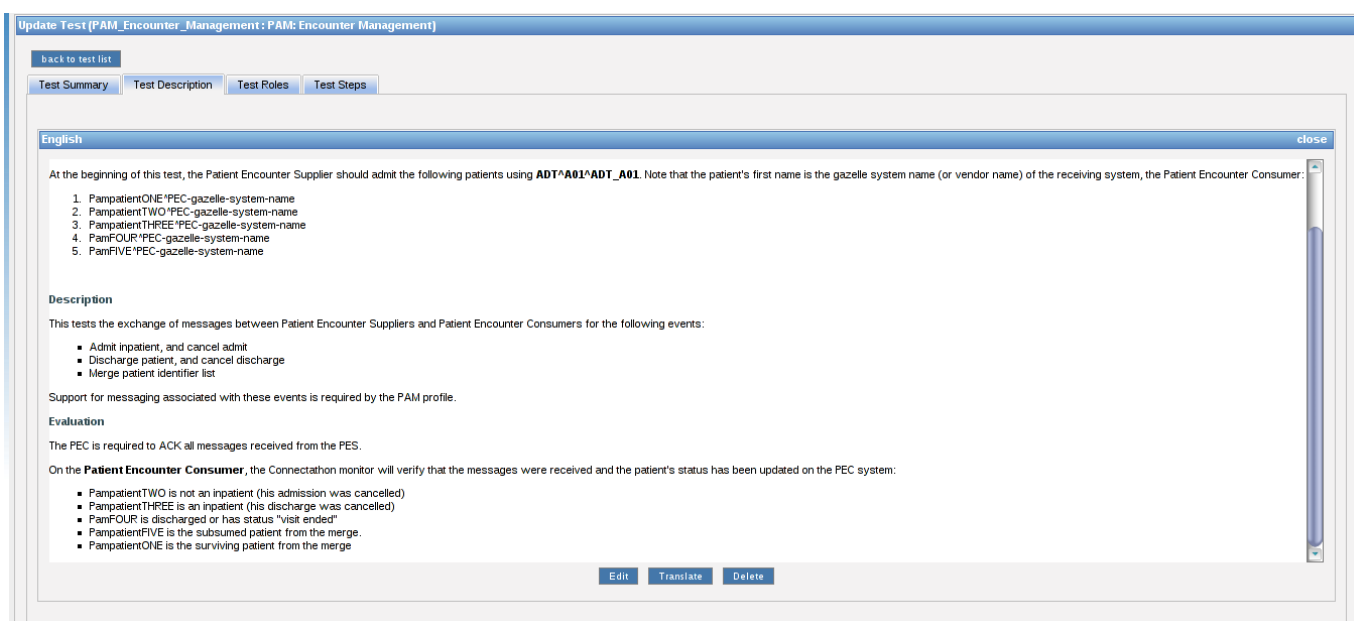
- Keyword, name and the short description are used to identify the test. By convention, the keyword name starts with the profile acronym.
- The test type is connectathon, pre-connectathon but can be other type as defined in the database.
- The test status indicates the readiness of the test. Only test with a status marked as ready are visible by the testers.
- The peer type indicates is the test is of type "No peer", "Peer to Peer" or "Group Test"
- The permanent link to the test is printed in this part (computed by Gazelle)
- The version of the test gives an indication of the most recent testing event for which the test was created/modified.

- The "Is orchestrable" attribute indicates whether the test will be run against a simulator (true) or against another system under test (false). When run against a simulator, the test requires the use of Bpel service and Gazelle web services to be orchestrated. Those services will enable the simulator to communicate with Gazelle in a standalone mode without any help from somebody.
- The "Is validated" attribute indicates whether the test is validated or not.

5.2 Test Description

This section describes very precisely the test scenario and gives indications to the vendor on how to perform the test, which tools are required and so on. This part also gives keys to the monitor about the things to check, the message to validate and so on. This part of the test can be translated into different languages. By convention, there are three sections in the test description:

- **Special Instruction:** contain information for the vendor of "special" considerations for this test, for example "ABC test must be run before this one", or "XYZ tool is used in this test"
- **Description:** Is a short overview of the scope of the test.
- **Evaluation:** These are the specific instructions to the connectathon monitor describing what evidence must be shown by the vendor in order to "pass" this test.



5.3 Test Roles

It is the most important part of the test, it is also the most complicated and confusing part of the work.

Assigning one or more Roles to a test determines which Actor/Integration Profile/Profile Option (AIPO) are involved in the test. Roles must be well-chosen for two reasons: (1) If a Role is assigned to a test, it means that the test will appear on the list of tests to do for any test system which supports the AIPO in the Role, and (2) only the transactions supported by the chosen Roles will be available when you define individual Test Steps on the next tab..

Prior to starting a test definition, you should ensure that the Test Roles you need for the test exist; if not, they can be created under **Tests Definition --> Role in test management**.

A test role (or role in test) is defined as a list of Actor/Integration profile/Profile Option and for each of these AIPO we must specify if the tuple is tested or not. The primary reason to include a Test Participant (ie an AIPO) in a Role with "Tested?" unchecked is because you want the transactions supported by that Test Participant (AIPO) to be used by the other test participants in that Role, but you do not want that test to show up as required for that test participant that is "not tested". This primarily occurs when one actor is "grouped" with another actor.

The whole test role can be set as "played by a tool", for example the OrderManager (formally RISMail) or the NIST registry or a simulator or so on.

A convention has been put in place for the naming of test roles:

<ACTOR_KEYWORD>_<INTEGRATION_PROFILE_KEYWORD>[_<PROFILE_OPTION_KEYWORD>][_ANY_OPTIONS][_WITH_SN][_WITH_ACTOR_KEYWORD][_HELPER]

If several actors from a profile or several profiles are used to defined the test role, only the main couple Actor/Integration Profile must be used to name the role.

- By ANY_OPTIONS we mean that any system implementing one of the option defined for the profile must perform the tests involving this role.
- _WITH_SN means that the transactions in which the role takes part must be run using TLS, consequently the involved actors must implement the Secure Node actor from ATNA profile. Note that, in that case, the Secure Node actor is set "not tested", so that failing this test do not fail the Secure Node actor.
- _WITH_ACTOR_KEYWORD means that the system must support a second actor, the one is not tested, in order to perform some initialization steps. For example PEC_PAM_WITH_PDC gathers the Patient Encounter Consumer actor from the Patient Admission Management profile and the Patient Demographic Consumer from the same profile; this is required because we need to populate the database of the PEC with some data received thanks to the PDC. Keep in mind that such associations must be meaningful that means that the gathered actors are linked by an IHE dependency.
- Finally, _HELPER means that the role is not tested but is required to ensure the coherence of the test.
- Here are some examples to let you better understand the naming convention:
- DOC_CONSUMER_XDS.b_ANY_OPTIONS gathers all the Document Consumer of the XDS.b profile no matter the options they support.
- IM_SWF_HELPER gathers all the Image Manager from the Schedule Workflow profile but those actors are not tested.

If the test participant is a tool or a simulator, we will used the system name as test role name: **<SIMULATOR or UTILITY_NAME>**, for instance ORDER_MANAGER, CENTRAL_ARCHIVE, NIST_REGISTRY and so on.

Tests Definition

Update Test (PAM_Encounter_Management : PAM: Encounter Management)

back to test list

Test Summary | Test Description | Test Roles | Test Steps

Keyword	# to realize	Card Min	Card Max	Optionality	URL	URL Doc	Action
PES_PAM_WITH_PDS	3	1	1	Required			edit delete
PEC_PAM_WITH_PDC	3	1	1	Required			edit delete

Add a role for test PAM_Encounter_Management

Keyword	#to realize	Card min	Card max	Optionality	URL	URL Documentation	Add
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Required	<input type="text"/>	<input type="text"/>	Add

Once you have chosen the roles involved in your test, you will be asked, for each of them to give some more information such as:

- **# to realize:** the number of times the system must realize with success this test for the tested actor to be validated. Typically, this is "3" for peer to peer tests, and "1" for No Peer and Group tests.
- **Card Min:** (cardinality) how many (at least) systems with this role must be involved in the test
- **Card Max:** (cardinality) how many (at most) systems with this role can be involved in the test
- **Optionality:** whether or not this role is a mandatory participant in the test. "Required" means a vendor cannot start the test until a system is identified for this role. "Optional" means that the test may be run whether or not a system is identified for this role in the test.

5.4 Test Steps

To help vendors with performing the test, we cut the test into small entities called test steps. In a newly defined test, when you first arrive on this page, you will find a sequence diagram only filled with the different roles you have previously defined. As you add test steps, you will be able to see the sequence diagram is automatically updated according to the steps you have defined. The red arrows stand for secured transaction (TLS set to true)

Test steps are ordered based on the step index, in most of the cases, vendors will have to respect the given order, especially if the test is run against a simulator.

Step Index	Initiator Role	Responder Role	Transaction	Secured	Message Type	Option	Description	Action																
10	PEC_PAM_WITH_PDC	PEC_PAM_WITH_PDC		<input type="checkbox"/>	NONE	Required	If possible, the PEC system should capture a copy of the ADT messages it receives from the PES. The Connection monitor may want to see them.	edit delete																
<p>Hide</p> <p>Input Contextual Informations</p> <table border="1"> <thead> <tr> <th>Id</th> <th>Label</th> <th>Path</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>Output Contextual Informations</p> <table border="1"> <thead> <tr> <th>Id</th> <th>Label</th> <th>Path</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>									Id	Label	Path	Value					Id	Label	Path	Value				
Id	Label	Path	Value																					
Id	Label	Path	Value																					
20	PES_PAM_WITH_PDS	PEC_PAM_WITH_PDC	ITI-31	<input type="checkbox"/>	ADT_A01	Required	ADT^A01^ADT_A01: Admit inpatients ONE, TWO, THREE, FOUR, FIVE: Prior to this message there should be an A28 if the patient identity did not already exist on the supplier side.	edit delete																
<p>Input and Output Contextual Information (0 - 0)</p>																								
30	PES_PAM_WITH_PDS	PEC_PAM_WITH_PDC	ITI-31	<input type="checkbox"/>	ADT_A09	Required	ADT^A11^ADT_A09: Cancel visit of patient TWO	edit delete																
<p>Input and Output Contextual Information (0 - 0)</p>																								
40	PES_PAM_WITH_PDS	PEC_PAM_WITH_PDC	ITI-31	<input type="checkbox"/>	ADT_A03	Required	ADT^A03^ADT_A03: Discharge patients THREE and FOUR	edit delete																
<p>Input and Output Contextual Information (0 - 0)</p>																								

Each step is described as follows:

- **Step index:** index of the step
- **Initiator role:** test role in charge of initiating the transaction (only roles with actors which are initiators for at least one transaction can be used as initiator role)
- **Responder role:** test role acting as the receiver for this step (only roles with actors which are responders for at least one transaction can be used as responder role)
- **Transaction:** the transaction to perform by the two actors
- **Secured:** indicates whether the transaction must be perform over TLS or not
- **Message type:** the message sent by the initiator
- **Option:** indicates whether the test step is required or not
- **Description:** detailed instructions on how to perform the step

When editing a step, you can choose to check or not the **Auto Response** box. When it is checked, it indicates that the selected role has to perform a step alone (initialization, log ...), no transaction nor message type have to be specified.

The screenshot shows the 'Tests Definition' web application. The main content area is titled 'Update Test (PAM_Encounter_Management : PAM_Encounter_Management)'. It features a navigation bar with 'back to test list' and tabs for 'Test Summary', 'Test Description', 'Test Roles', and 'Test Steps'. The 'Test Description' tab is active. The form contains the following fields:

- Step Index: 30
- Secured:
- Auto Response:
- Initiator Role: PES_PAM_WITH_PDS
- Transaction: ITI-31
- Responder Role: PEC_PAM_WITH_PDC
- Message Type: ADT_A09
- Option: Required
- Description: ADT_A09: Cancel visit of patient TWO

At the bottom, there are buttons for 'Contextual Information Input', 'Contextual Information Output', 'Cancel', and 'Update this step'.

In order not to waste time editing steps for a little change, the step index field, secured checkbox, option selection and description fields can be filled from the main page of test steps. The change is recorded in database each time you lose the focus of the modified field.

If you have chosen to write an orchestrated test, that means that the system under test will communicate with a simulator, you may have to enter some more informations called "Contextual Information". In some cases, those informations are needed by the simulator to build a message which match the system configuration or data. This can be used to specify a patient ID known by the system under test for instance.

Two kinds of contextual informations are defined:

- **Input Contextual Information:** The information provided to the simulator

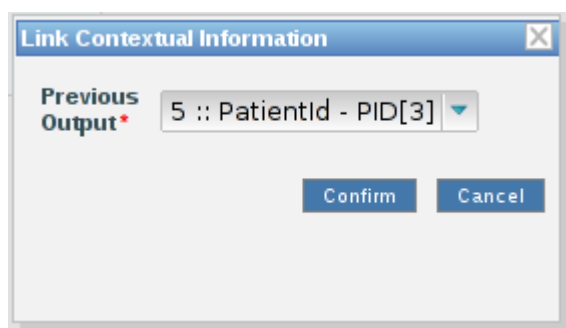
- **Output Contextual Information:** The information sent back by the simulator and that can be used as input contextual information for the next steps

For each contextual information, you are expected to provide the label of the field and the path (it can be XPath or HL7 path if you need to feed a specific XML element or HL7v2 message segment). A default value can also be set.



The screenshot shows a dialog box titled "Add Contextual Information". It has three input fields: "Label *", "Path *", and "Value". Below the fields are two buttons: "Confirm" and "Cancel".

If you have defined output contextual informations for previous steps, you can use them as input contextual information for next steps by importing them, as it is shown on the capture below. So that, the simulator will received the return of a previous step as new information and will be able to build next messages.



The screenshot shows a dialog box titled "Link Contextual Information". It has a dropdown menu for "Previous Output*" with the selected value "5 :: PatientId - PID[3]". Below the dropdown are two buttons: "Confirm" and "Cancel".

For more details about the expectation of simulators, read the developer manual of the simulator you want to involve in your test. A short example based on XCA Initiating Gateway Simulator use is given below.

XCA Initiating Gateway supports two transactions: ITI-38 for querying the responding gateway about the documents for a specific patient and ITI-39 to retrieve those documents. In a first step we may ask the responding gateway for the documents of patient 1234^^^&1.2.3.4.5.6&ISO, in the second step we will ask the responding gateway to send the first retrieved document.

	label	path	value
step 1	<i>Input Contextual Information</i>		
	XDSDocumentEntryPatientId	\$XDSDocumentEntry.patientId	1234^&1.2.3.4.5.6&ISO
	<i>Output Contextual Information</i>		
	XDSDocumentEntryUniqueId	\$XDSDocumentEntry.uniqueId	7.9.0.1.2.3.4
step 2	<i>Input Contextual Information</i>		
	XDSDocumentEntryUniqueId	\$XDSDocumentEntry.uniqueId	7.9.0.1.2.3.4

In this way, no action on the simulator side is required from the vendor, he/she only has to set up his/her system under test and give the first input contextual information to the simulator through the Test Management user interface.

6 Meta tests

In some Peer to Peer tests, the transactions supported by one Role are identical across multiple different tests, yet that Role's partners across those tests are different. This is best illustrated by an example: In Cardiology and Radiology workflow profiles, a scheduling system (Order Filler Role) profiles a worklist to various imaging systems (Modality Roles). A vendors' Order Filler may play the Order Filler Role in Radiology SWF profile and Cardiology ECHO, CATH and STRESS profiles. If the Order Filler may be assigned a Peer to Peer "worklist" test with modalities in each of these profiles. This could result in 12 worklist tests to pass for the Order Filler (3 worklist tests x 4 profiles). **Meta Tests** allow test definers to eliminate this kind of redundant testing.

Meta tests are special tests are built of equivalent test definitions for a given test role. Actually, we try not to duplicate tests but it can happen that two different tests are the same according the point of view of one test role involved in both. In that case, we merge the two tests under one Meta Test for this specific role.

When a vendor sees a Meta Test in his/her system's test list the equivalent tests are listed within the meta test. He/she is allowed to perform 3 instances of any of the tests within the meta test instead of three instances for each individual test.. That means that if the meta test is composed of 4 tests, the involved actor is expected to have any combination of 3 instances verified,

Meta tests are defined in gazelle under **Test Definition --> Meta test list**. A Meta test is given a keyword and a short description; then the equivalent tests are linked to the meta test.

As an example, let's take the meta test with keyword `Meta_Doc_Repository_Load`. This Meta test gathers four tests defined, among other, for the Document Repository actor of the XDS-I.b profile. Each of these tests ask this actor to perform the RAD-68 and ITI-42 transactions against an actor supporting several options. From the point of view of the Document Repository, those four tests are equivalent since we are testing four times the same transactions. Consequently, running only three of the twelve instances it would have had to do is enough to be successfully graded.

7 Configuration of master model slaves

This page provides the instructions on how to add a slave application to the master model.

7.1 Pre-requisite

- **slony 2.0.6** : The version of slony on the slave and on the master shall be identical. Currently the version in use is 2.0.6. Run the following command to find out the version you are running

```
1 admin@master:~$ slon -v
2 slon version 2.0.6
3 admin@master:~$
4 admin@slave:~$ slon -v
5 slon version 2.0.6
6 admin@slave:~$
```

- **postgresql 8.4.x** : the version of postgresql on the slave and on the master are not required to be identical. The version used shall be one of the 8.4 series. The master is running 8.4.7. The master system needs to access the database on the slave. This achieved by configuring the file `pg_hba.conf` on the **slave**.
- Make sure that the file `pg_hba.conf` on the slave contains the following entries.

```
1 # TYPE DATABASE USER CIDR-ADDRESS METHOD
2 host gazelle-on-slave gazelle 131.254.209.12/32 md5
3 host gazelle-on-slave gazelle 131.254.209.13/32 md5
4 host gazelle-on-slave gazelle 131.254.209.14/32 md5
5 host gazelle-on-slave gazelle 131.254.209.15/32 md5
```

where `host` is the `gazelle-on-slave` is the name of the gazelle database on the slave. When the configuration of the slave is succesful then you should be able to run the following command

```
1 psql -h slave -U username gazelle-on-slave
```

and access the remote database.

One this level of configuration is reach we can start configuring slony on the master and on the slave.

7.2 Initialisation of slony on the master system

I usually have them in `~/slony`

The slony initialisation script is stored in the file : `slonik_init.sk`. The file should be executable. When this script is run it creates a new schema on each of the nodes (slaves and master). If you need to rerun the script, make sure that you delete the schema from each of the nodes

```
1 DROP SCHEMA "_TF" CASCADE ;
```

- Content of the file : `slonik_init.sk`

```
#!/usr/bin/slony
define CLUSTER TF;
define PRIMARY 1;
define EPSOS 10;
```

```
define TM 20;
define PR 30;
define ORANGE 60;
cluster name = @CLUSTER;
# Here we declare how to access each of the nodes. Master is PRIMARY and others
are the slaves.node @PRIMARY admin conninfo = 'dbname=master-model
host=jumbo.irisa.fr user=gazelle password=XXXXXX';
node @TM admin conninfo = 'dbname=ihe-europe-2010 host=kujira.irisa.fr
user=gazelle password=XXXXXX';
node @PR admin conninfo = 'dbname=product-registry host=jumbo.irisa.fr
user=gazelle password=XXXXXX';
node @EVSCLIENT admin conninfo = 'dbname=evs-client-prod host=jumbo.irisa.fr
user=gazelle password=XXXXXX';
node @ORANGE admin conninfo = 'dbname=gazelle-na-2012 host=gazelle-
orange.wustl.edu user=gazelle password=XXXXXX'
# Initialisation of the cluster
init cluster (id=@PRIMARY, comment='Gazelle Master Model');
# Declaration of the slaves
store node (id=@TM, event node=@PRIMARY, comment='Test Management Slave');
store node (id=@PR, event node=@PRIMARY, comment='Product Registry Slave');
store node (id=@EVSCLIENT, event node=@PRIMARY, comment='EVS Client Slave');
store node (id=@ORANGE, event node=@PRIMARY, comment='Test Management Slave
Orange');
# Define the path from Slaves to Master
store path (server=@PRIMARY, client=@TM, conninfo='dbname=master-model
host=jumbo.irisa.fr user=gazelle');
store path (server=@PRIMARY, client=@PR, conninfo='dbname=master-model
host=jumbo.irisa.fr user=gazelle');
store path (server=@PRIMARY, client=@EVSCLIENT, conninfo='dbname=master-model
host=jumbo.irisa.fr user=gazelle');
store path (server=@PRIMARY, client=@ORANGE, conninfo='dbname=master-model
host=jumbo.irisa.fr user=gazelle');
# Define the path from Master to Slaves
store path (server=@TM, client=@PRIMARY, conninfo='dbname=ihe-europe-2010
host=kujira.irisa.fr user=gazelle');
store path (server=@PR, client=@PRIMARY, conninfo='dbname=product-registry
host=jumbo.irisa.fr user=gazelle');
store path (server=@EVSCLIENT, client=@PRIMARY, conninfo='dbname=evs-client-prod
host=jumbo.irisa.fr user=gazelle');
store path (server=@ORANGE, client=@PRIMARY, conninfo='dbname=gazelle-na-2012
host=gazelle-orange.wustl.edu user=gazelle password=gazelle');
```

■ The next file to consider is : script_server.sk

```
#!/usr/bin/slonik
define CLUSTER TF;
define PRIMARY 1;
define TM 20;
define PR 30;
define EVSCLIENT 40;
define ORANGE 60;
cluster name = @CLUSTER;
#Declaration of the nodes
node @PRIMARY admin conninfo = 'dbname=master-model host=jumbo.irisa.fr
```

```
user=gazelle password=gazelle';node @TM admin conninfo = 'dbname=ihe-europe-2010
host=kujira.irisa.fr user=gazelle password=gazelle';
node @PR admin conninfo = 'dbname=product-registry host=jumbo.irisa.fr
user=gazelle password=gazelle';
node @EVSCCLIENT admin conninfo = 'dbname=evs-client-prod host=jumbo.irisa.fr
user=gazelle password=gazelle';
node @ORANGE admin conninfo = 'dbname=gazelle-na-2012 host=gazelle-
orange.wustl.edu user=gazelle password=gazelle';
# We need 2 sets: One for the Technical Framework (TF) part and one for the Test
Definition (Test Management = TM) part
create set (id=1, origin=@PRIMARY, comment='TF');
create set (id=2, origin=@PRIMARY, comment='TM');
# Assign the table and sequences to each of the nodes
set add table (id=176, set id=1, origin = @PRIMARY, fully qualified name =
'public.revinfo', comment = 'table');
set add table (id=174, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor', comment = 'table');
set add table (id=175, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor_aud', comment = 'table');
set add sequence (id=2, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor_id_seq', comment = 'seq');
set add table (id=3, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_hl7_message_profile_table', comment = 'table');
set add table (id=4, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_hl7_message_profile_table_aud', comment = 'table');
set add sequence (id=5, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_hl7_message_profile_table_id_seq', comment = 'seq');
set add table (id=6, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor_integration_profile_option', comment = 'table');
set add table (id=7, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor_integration_profile_option_aud', comment = 'table');
set add sequence (id=8, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor_integration_profile_option_id_seq', comment = 'seq');
set add table (id=9, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile', comment = 'table');
set add table (id=10, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_aud', comment = 'table');
set add sequence (id=11, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_id_seq', comment = 'seq');
set add table (id=12, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_option', comment = 'table');
set add table (id=13, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_option_aud', comment = 'table');
set add sequence (id=14, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_option_id_seq', comment = 'seq');
set add table (id=15, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor_integration_profile', comment = 'table');
set add table (id=16, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor_integration_profile_aud', comment = 'table');
set add sequence (id=17, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_actor_integration_profile_id_seq', comment = 'seq');
set add table (id=18, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_type', comment = 'table');
set add table (id=19, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_type_aud', comment = 'table');
set add sequence (id=20, set id=1, origin = @PRIMARY, fully qualified name =
```

```
'public.tf_integration_profile_type_id_seq', comment = 'seq');
set add table (id=21, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_link', comment = 'table');
set add table (id=22, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_link_aud', comment = 'table');
set add sequence (id=23, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_link_id_seq', comment = 'seq');
set add table (id=24, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_option_type', comment = 'table');
set add table (id=25, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_option_type_aud', comment = 'table');
set add sequence (id=26, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_option_type_id_seq', comment = 'seq');
set add table (id=27, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_profile_link', comment = 'table');
set add table (id=28, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_profile_link_aud', comment = 'table');
set add sequence (id=29, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_profile_link_id_seq', comment = 'seq');
set add table (id=30, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_status_type', comment = 'table');
set add table (id=31, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_status_type_aud', comment = 'table');
set add sequence (id=32, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_status_type_id_seq', comment = 'seq');
set add table (id=33, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_profile_inria_hl7_validation_files', comment = 'table');
set add table (id=34, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_profile_inria_hl7_validation_files_aud', comment = 'table');
set add sequence (id=35, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_profile_inria_hl7_validation_files_id_seq', comment = 'seq');
set add table (id=36, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_configuration_mapped_with_aipo', comment = 'table');
set add table (id=37, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_configuration_mapped_with_aipo_aud', comment = 'table');
set add sequence (id=38, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_configuration_mapped_with_aipo_id_seq', comment = 'seq');
set add table (id=39, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_conf_mapping_w_aipo_w_conftypes', comment = 'table');
set add table (id=40, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_conf_mapping_w_aipo_w_conftypes_aud', comment = 'table');
set add table (id=42, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_conftype_w_ports_wstype_and_sop_class', comment = 'table');
set add table (id=43, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_conftype_w_ports_wstype_and_sop_class_aud', comment = 'table');
set add sequence (id=44, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_conftype_w_ports_wstype_and_sop_class_id_seq', comment = 'seq');
set add table (id=45, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_status_type', comment = 'table');
set add table (id=46, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_status_type_aud', comment = 'table');
set add sequence (id=47, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_status_type_id_seq', comment = 'seq');
set add table (id=48, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_configuration_type', comment = 'table');
set add table (id=49, set id=2, origin = @PRIMARY, fully qualified name =
```

```
'public.tm_configuration_type_aud', comment = 'table');
set add sequence (id=50, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_configuration_type_id_seq', comment = 'seq');
set add table (id=51, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_meta_test', comment = 'table');
set add table (id=52, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_meta_test_aud', comment = 'table');
set add sequence (id=53, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_meta_test_id_seq', comment = 'seq');
set add table (id=54, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_meta_test_test_roles', comment = 'table');
set add table (id=55, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_meta_test_test_roles_aud', comment = 'table');
set add table (id=57, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_role_in_test_test_participants', comment = 'table');
set add table (id=58, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_role_in_test_test_participants_aud', comment = 'table');
set add table (id=60, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_sop_class', comment = 'table');
set add table (id=61, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_sop_class_aud', comment = 'table');
set add sequence (id=62, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_sop_class_id_seq', comment = 'seq');
set add table (id=63, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_description', comment = 'table');
set add table (id=64, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_description_aud', comment = 'table');
set add sequence (id=65, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_description_id_seq', comment = 'seq');
set add table (id=66, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_participants', comment = 'table');
set add table (id=67, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_participants_aud', comment = 'table');
set add sequence (id=68, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_participants_id_seq', comment = 'seq');
set add table (id=69, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_test_description', comment = 'table');
set add table (id=70, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_test_description_aud', comment = 'table');
#set add sequence (id=71, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_test_description_id_seq', comment = 'seq');
set add table (id=72, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_input_ci', comment = 'table');
set add table (id=73, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_input_ci_aud', comment = 'table');
set add table (id=75, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_option', comment = 'table');
set add table (id=76, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_option_aud', comment = 'table');
set add sequence (id=77, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_option_id_seq', comment = 'seq');
set add table (id=78, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_output_ci', comment = 'table');
set add table (id=79, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_output_ci_aud', comment = 'table');
set add table (id=81, set id=2, origin = @PRIMARY, fully qualified name =
```



```
'public.tm_test_steps', comment = 'table');
set add table (id=82, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_aud', comment = 'table');
set add sequence (id=83, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_steps_id_seq', comment = 'seq');
set add table (id=84, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_status', comment = 'table');
set add table (id=85, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_status_aud', comment = 'table');
set add sequence (id=86, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_status_id_seq', comment = 'seq');
set add table (id=87, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_roles', comment = 'table');
set add table (id=88, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_roles_aud', comment = 'table');
set add sequence (id=89, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_roles_id_seq', comment = 'seq');
set add table (id=90, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_web_service_type', comment = 'table');
set add table (id=91, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_web_service_type_aud', comment = 'table');
set add sequence (id=92, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_web_service_type_id_seq', comment = 'seq');
set add table (id=93, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_type', comment = 'table');
set add table (id=94, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_type_aud', comment = 'table');
set add sequence (id=95, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_type_id_seq', comment = 'seq');
set add table (id=96, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_domain', comment = 'table');
set add table (id=97, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_domain_aud', comment = 'table');
set add sequence (id=98, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_domain_id_seq', comment = 'seq');
set add table (id=99, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_type_link', comment = 'table');
set add table (id=100, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_integration_profile_type_link_aud', comment = 'table');
set add table (id=102, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_test_steps', comment = 'table');
set add table (id=103, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_test_steps_aud', comment = 'table');set add table (id=105, set
id=2, origin = @PRIMARY, fully qualified name = 'public.tm_test_option', comment
= 'table');
set add table (id=106, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_option_aud', comment = 'table');
set add sequence (id=107, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_option_id_seq', comment = 'seq');
set add table (id=108, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_peer_type', comment = 'table');
set add table (id=109, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_peer_type_aud', comment = 'table');
set add sequence (id=110, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_peer_type_id_seq', comment = 'seq');
set add table (id=111, set id=1, origin = @PRIMARY, fully qualified name =
```

```
'public.tf_hl7_message_profile', comment = 'table');
set add table (id=112, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_hl7_message_profile_aud', comment = 'table');
set add sequence (id=113, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_hl7_message_profile_id_seq', comment = 'seq');
set add table (id=114, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_domain_profile', comment = 'table');
set add table (id=115, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_domain_profile_aud', comment = 'table');
set add table (id=117, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction', comment = 'table');
set add table (id=118, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_aud', comment = 'table');
set add sequence (id=119, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_transaction_id_seq', comment = 'seq');
set add table (id=177, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_hl7_message_profile_affinity_domain', comment = 'table');
set add table (id=178, set id=1, origin = @PRIMARY, fully qualified name =
'public.tf_hl7_message_profile_affinity_domain_aud', comment = 'table');
set add table (id=120, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test', comment = 'table');
set add table (id=121, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_aud', comment = 'table');
set add sequence (id=122, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_test_id_seq', comment = 'seq');
set add table (id=123, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_transport_layer_for_config', comment = 'table');
set add table (id=124, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_transport_layer_for_config_aud', comment = 'table');
set add sequence (id=125, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_transport_layer_for_config_seq', comment = 'seq');
set add table (id=126, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_role_in_test', comment = 'table');
set add table (id=127, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_role_in_test_aud', comment = 'table');
set add sequence (id=128, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_role_in_test_id_seq', comment = 'seq');
set add table (id=129, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_contextual_information', comment = 'table');
set add table (id=130, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_contextual_information_aud', comment = 'table');
set add sequence (id=131, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_contextual_information_id_seq', comment = 'seq');
set add table (id=132, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_file', comment = 'table');
set add table (id=133, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_file_aud', comment = 'table');
set add sequence (id=134, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_file_id_seq', comment = 'seq');
set add table (id=135, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_creator', comment = 'table');
set add table (id=136, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_creator_aud', comment = 'table');
set add sequence (id=137, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_creator_id_seq', comment = 'seq');
set add table (id=138, set id=2, origin = @PRIMARY, fully qualified name =
```

```
'public.tm_object_attribute', comment = 'table');
set add table (id=139, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_attribute_aud', comment = 'table');
set add sequence (id=140, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_attribute_id_seq', comment = 'seq');
set add table (id=141, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_file_method', comment = 'table');
set add table (id=142, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_file_method_aud', comment = 'table');
set add table (id=144, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_attribute_option', comment = 'table');
set add table (id=145, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_attribute_option_aud', comment = 'table');
set add sequence (id=146, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_attribute_option_id_seq', comment = 'seq');
set add table (id=147, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_class_validator', comment = 'table');
set add table (id=148, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_class_validator_aud', comment = 'table');
set add sequence (id=149, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_class_validator_id_seq', comment = 'seq');
set add table (id=150, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_file_type', comment = 'table');
set add table (id=151, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_file_type_aud', comment = 'table');
set add sequence (id=152, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_file_type_id_seq', comment = 'seq');
set add table (id=153, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_instance_validation', comment = 'table');
set add table (id=154, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_instance_validation_aud', comment = 'table');
set add sequence (id=155, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_instance_validation_id_seq', comment = 'seq');
set add table (id=156, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_method_parameter', comment = 'table');
set add table (id=157, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_method_parameter_aud', comment = 'table');
set add table (id=159, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_parameter_validator', comment = 'table');
set add table (id=160, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_parameter_validator_aud', comment = 'table');
set add sequence (id=161, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_parameter_validator_id_seq', comment = 'seq');
set add table (id=162, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_reader', comment = 'table');
set add table (id=163, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_reader_aud', comment = 'table');
set add sequence (id=164, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_reader_id_seq', comment = 'seq');
set add table (id=165, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_type_status', comment = 'table');
set add table (id=166, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_type_status_aud', comment = 'table');
set add sequence (id=167, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_type_status_id_seq', comment = 'seq');
set add table (id=168, set id=2, origin = @PRIMARY, fully qualified name =
```

```
'public.tm_object_type', comment = 'table');
set add table (id=169, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_type_aud', comment = 'table');
set add sequence (id=170, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_type_id_seq', comment = 'seq');
set add table (id=171, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_method_validator', comment = 'table');
set add table (id=172, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_method_validator_aud', comment = 'table');
set add sequence (id=173, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_object_method_validator_id_seq', comment = 'seq');
set add table (id=179, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_path', comment = 'table');
set add table (id=180, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_path_aud', comment = 'table');
set add sequence (id=181, set id=2, origin = @PRIMARY, fully qualified name =
'public.tm_path_id_seq', comment = 'seq');
set add table (id=182, set id=2, origin = @PRIMARY, fully qualified name =
'public.tf_ws_transaction_usage', comment = 'table');
set add table (id=183, set id=2, origin = @PRIMARY, fully qualified name =
'public.tf_ws_transaction_usage_aud', comment = 'table');
set add sequence (id=184, set id=2, origin = @PRIMARY, fully qualified name =
'public.tf_ws_transaction_usage_id_seq', comment = 'seq');
# Then for each slave we tell to start the sync
#TM
subscribe set (id = 1, provider = @PRIMARY, receiver = @TM);
sync(id=@PRIMARY);
wait for event(origin=@PRIMARY, confirmed=@TM, wait on=@PRIMARY);
subscribe set (id = 2, provider = @PRIMARY, receiver = @TM);
sync(id=@PRIMARY);
wait for event(origin=@PRIMARY, confirmed=@TM, wait on=@PRIMARY);
#EVSCLIENT
subscribe set (id = 1, provider = @PRIMARY, receiver = @EVSCLIENT);
sync(id=@PRIMARY);
wait for event(origin=@PRIMARY, confirmed=@EVSCLIENT, wait on=@PRIMARY);
#PR
subscribe set (id = 1, provider = @PRIMARY, receiver = @PR);
sync(id=@PRIMARY);
wait for event(origin=@PRIMARY, confirmed=@PR, wait on=@PRIMARY);
#ORANGE
subscribe set (id = 1, provider = @PRIMARY, receiver = @ORANGE);
sync(id=@PRIMARY);
wait for event(origin=@PRIMARY, confirmed=@ORANGE, wait on=@PRIMARY);
subscribe set (id = 2, provider = @PRIMARY, receiver = @ORANGE);
sync(id=@PRIMARY);
wait for event(origin=@PRIMARY, confirmed=@ORANGE, wait on=@PRIMARY);
```

7.3 Initialization of slony on the slaves

Starting the slon process is not an easy command to type, so I have made a script on each of the slaves to execute the command.

```
1 nohup slon TF "dbname=evs-client-prod user=gazelle" > evs-client-prod.log
```

7.4 Launching the synchronization or restarting the synchronization

If you are launching the synchronization for the first time (seen from the master) then you can start from point 4. At any point in the process if you encounter an error, you will need to restart from 1.

1. kill the slon process on all **slaves** : `killall slon`
2. kill the slon process on the **master** : `killall slon`
3. drop the "_TF" schema in all **slaves** databases
4. on the **master** run the `slonik_init.sk` script.
5. on each of the **slaves** start the slon processes
6. on the **master** start the slon process
7. on the **master** run the `script_server.sk` script
8. check the log files on each of the slaves and the master in order to make sure that the synchronization is actually taking place.