

**Program Guide Minor Ailments**

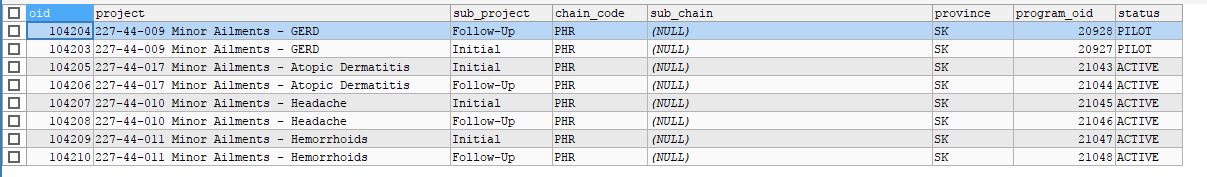
*This document is not to be released to a third party without the consent of Pharmacy Access Solutions Inc. It is provided to you in confidence and contains information which, if released, will be harmful to the competitive position of Pharmacy Access Solutions Inc.*

The minor ailments program is an on demand simple survey type of program. Here is a brief example of how the user would normally use the program before configuration is explained:

|  |  |
| --- | --- |
| Select the [PROFESSIONAL SERVICES] button from the applicable patient’s PATIENT VIEW. | Select “Minor Ailments”. Select [LAUNCH]. |
| A full list of Minor Ailments options will appear. Select “Atopic Dermatitis – Initial Assessment” or “Atopic Dermatitis– Follow-up Assessment” and select [LAUNCH]. | |
| The applicable Atopic Dermatitis Documentation Panel will be placed in the Open Interventions and Interventions tabs of the *Interventions* facility and PATIENT VIEW. | |

The minor ailments project is made up of many different sub programs. Each ailment usually consists of 2 parts, an initial assessment and a follow up. Each part is made up of 1 program oid in our database.

Example:



The first part of every program starts with the program table:

For example if we were to look at Atopic Dermatitis we would be looking at program oids 21043 and 21044.

Setup for the 2 programs would look like this in the program table:



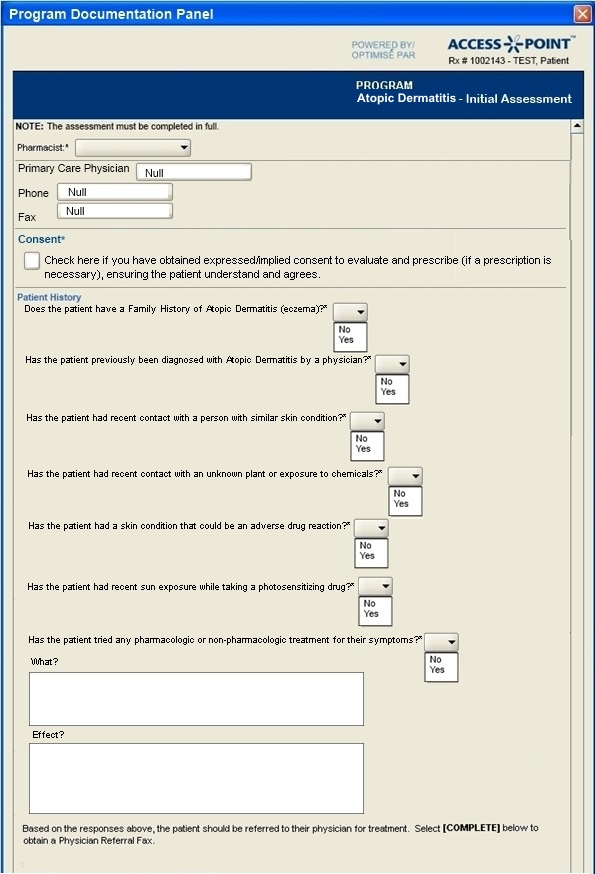
As you can see there are many options that can be configured on the program table.

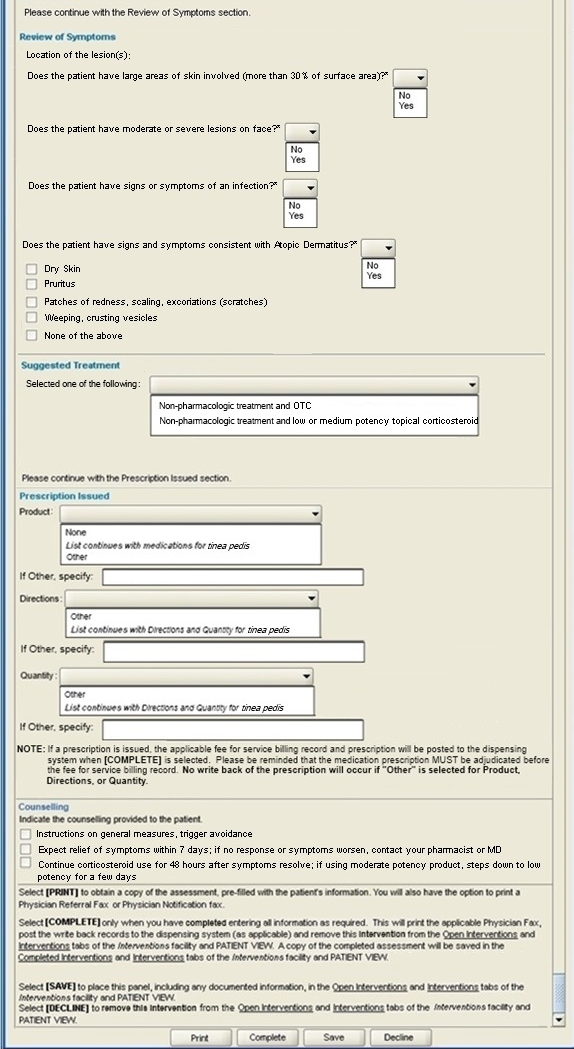
|  |  |
| --- | --- |
| **Column name** | **DESCRIPTION** |
|
| Oid | Unique value. One of the most referenced values that ties the other configuration tables back to a program. |
| Name | Controls how the program name appears to the user both on launch panel and in program listings for patient: |
| allow\_multiple\_registrations | If set to 1, the patient can be registered into the program more than once. |
| allow\_multiple\_declines | If set to 1, the program can be declined multiple times. It only makes sense to set this to 1 if the allo\_multiple\_registrations flag is also set to 1. |
| Patient\_centric | It set to 0, then the program will be prompted using the din. |
| otc\_program | It set to 1, program will be shown in the Dashboard Programs/Resources Facility |
| Expiry\_days | If value is set to this field. The registration will expire based on the number provided in this field. |
| Program\_group | This field is used to group 2 or more programs. When program oid is provided, only 1 of group program will trigger |
| max\_opportunities\_per\_day | If set to 1, the output will be receive once a day by patient. |
| Monitor\_patient\_on\_registration | If set to 1, all succeeding fills of registered patient will be monitored |
| category\_type | If set to **Null**, the program will be triggered thru prompting. If set to ‘**PPS-WD**’, the program will be shown in the Professional Services, without highlighting a drug. If set to **‘PPS-D’**, the program will be shown in the Professional Services, when highlighting a drug. If set to **‘CE’**, the program will be shown in the Resources Facility  CE module tab. If set to ‘**DM**’, the program will be shown in the Resources Facility  DEMO Meter tab. If set to **‘PPS-MD’**, the program will be shown in the Professional Services, when highlighting multiple drugs. If set to ‘**PRO\_INFO**’, the program will be shown in the Resources Facility  Program Information tab. If set to ‘**PS**’, the program will be shown in the Resources Facility  Pharmacist Information tab. If set to ‘**REC**’, the program will be shown in the Resources Facility  Patient Tools tab. |
| Action config | Used to control how a program functions when triggered. This is how minor ailments programs are set up:   |  |  | | --- | --- | |  | --- - !Action event: INIT\_WINDOW action: SHOW\_BUTTON value: "PRINT,COMPLETE,DEFER,DECLINE" - !Action event: NEW\_PROGRAM action: OPEN value: DEFER\_ONLY - !Action event: SUBMIT\_REGISTRATION action: CREATE\_APPOINTMENT value: YES |   There are 3 different things that are configured here.   1. The first action determines which buttons will be shown to the user. 2. Second action details when the program is triggered what will happen. Some programs can be set to print automatically, pop up, etc 3. The third action defines that when the survey is completed and submitted the add appointment panel will display. |
| Drug company oid | Used to help group programs under a specific umbrella. Used when a program has drugs assigned to it so individual programs can have the same drugs but could display those drugs differently. |
| Sequence | Used for when multiple programs are under the same drug company |
| Survey oid | \*\* This value is important for minor ailments since this type of program relies mainly on a survey |
| Program type | Minor ailments are under the SIMPLE SURVEY type. Simple survey type programs do not require a additional consent in order to complete and do not submit all details like patient data. |
|  |  |

The supporting tables for programs are as follows:

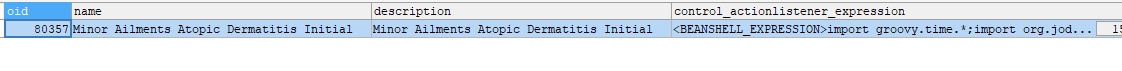
**Survey**

Here is how the program appears to a user after launch:





This is how a survey will appear to a user. Each item in this survey is made of an element. The survey is the main entry which elements reference. However there is one field in the survey table which is very important to the minor ailments program. This is the control\_actionlistener\_experssion.



The control action listener expression references survey elements and can do different things to elements. The main thing that this program uses the action listener for is to turn elements off and on depending what actions are taken. Note that this gets evaluated after each action on the survey so important care is required to not create too much overhead or it will impact performance.

Example action listener:

<BEANSHELL\_EXPRESSION>

import groovy.time.\*;

import org.joda.time.\*;

import org.joda.time.format.\*;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import com.pasi.script.groovy.DirectiveProcessor;

import com.pasi.directive.DirectiveHelper;

import com.pasi.gui.SimpleSurvey.FrameSimpleSurvey;

import com.pasi.common.AgentDirective;

import java.math.\*;

import com.pasi.survey.AgentSurveyScriptHelper;

import com.pasi.gui.hcdpd.HCDPDApi;

import com.pasi.agentdb.domain.hcdpd.\*;

import java.util.List;

import javax.swing.JComponent;

try {

// This stops the program from creating an appointment when submit is clicked

\_directiveMap.put("SUBMIT\_REGISTRATION","CREATE\_APPOINTMENT=NO")

Below is the main type of entry used to turn off an on elements. The first field references against the element id used in a survey element to control if it is visible or not. False= not visible. True=visible

\_surveyPanel.setSurveyElementControlVisibility("medical\_history\_specify", false);

\_surveyPanel.setSurveyElementControlVisibility("medical\_allergies\_specify", false);

\_surveyPanel.setSurveyElementControlVisibility("skin\_condition\_recommendation", false);

//\_surveyPanel.setSurveyElementControlVisibility("chemical\_exposure", false);

\_surveyPanel.setSurveyElementControlVisibility("exposure\_recommendation", false);

//\_surveyPanel.setSurveyElementControlVisibility("adverse\_drug\_reaction", false);

\_surveyPanel.setSurveyElementControlVisibility("adverse\_recommendation", false);

//\_surveyPanel.setSurveyElementControlVisibility("photosensitizing\_drug", false);

\_surveyPanel.setSurveyElementControlVisibility("photosentisizing\_recommendation", false);

//\_surveyPanel.setSurveyElementControlVisibility("diagnosed\_headache", false);

\_surveyPanel.setSurveyElementControlVisibility("pharmacologic\_what", false);

\_surveyPanel.setSurveyElementControlVisibility("pharmacologic\_effect", false);

\_surveyPanel.setSurveyElementControlVisibility("yes\_response\_logic", false);

\_surveyPanel.setSurveyElementControlVisibility("no\_response\_logic", false);

\_surveyPanel.setSurveyElementControlVisibility("label5", false);

\_surveyPanel.setSurveyElementControlVisibility("consistent\_atopic\_dermatitis", false);

\_surveyPanel.setSurveyElementControlVisibility("symptoms\_dermatitis", false);

\_surveyPanel.setSurveyElementControlVisibility("symptoms\_dermatitis\_note", false);

\_surveyPanel.setSurveyElementControlVisibility("lesions\_location", false);

\_surveyPanel.setSurveyElementControlVisibility("large\_skin\_involved", false);

\_surveyPanel.setSurveyElementControlVisibility("large\_skin\_recommendation", false);

\_surveyPanel.setSurveyElementControlVisibility("lesions\_face", false);

\_surveyPanel.setSurveyElementControlVisibility("moderate\_recommendation", false);

\_surveyPanel.setSurveyElementControlVisibility("infection\_symptoms", false);

\_surveyPanel.setSurveyElementControlVisibility("infection\_recommendation", false);

\_surveyPanel.setSurveyElementControlVisibility("yes\_symptoms\_logic", false);

\_surveyPanel.setSurveyElementControlVisibility("no\_symptoms\_logic", false);

\_surveyPanel.setSurveyElementControlVisibility("label\_treatment", false);

\_surveyPanel.setSurveyElementControlVisibility("recommended\_treatment", false);

\_surveyPanel.setSurveyElementControlVisibility("infant\_recommendation", false);

\_surveyPanel.setSurveyElementControlVisibility("older\_child\_recommendation", false);

\_surveyPanel.setSurveyElementControlVisibility("older\_child\_recommendation2", false);

\_surveyPanel.setSurveyElementControlVisibility("label6", false);

\_surveyPanel.setSurveyElementControlVisibility("issued\_prescription", false);

\_surveyPanel.setSurveyElementControlVisibility("text\_footer14", false);

\_surveyPanel.setSurveyElementControlVisibility("counseling", false);

//\_surveyPanel.setSurveyElementControlVisibility("label1", false);

//\_surveyPanel.setSurveyElementControlVisibility("follow\_up\_date", false);

//\_surveyPanel.setSurveyElementControlVisibility("follow\_up\_location", false);

//\_surveyPanel.setSurveyElementControlVisibility("follow\_up\_label", false);

\_surveyPanel.setSurveyElementControlVisibility("phone\_specify", false);

// prescription elements

\_surveyPanel.setSurveyElementControlVisibility("rationale\_prescribing", false);

\_surveyPanel.setSurveyElementControlVisibility("medSask\_text", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_textsearch", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_search", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_drug", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_strength", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_din", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_sig", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_qty", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_orig\_fill", false);

\_surveyPanel.setSurveyElementControlVisibility("text\_refills", false);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_rx\_date", false);

if ("Telephone".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("follow\_up\_location"))) {

\_surveyPanel.setSurveyElementControlVisibility("phone\_specify", true);

}

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("medical\_history"))) {

\_surveyPanel.setSurveyElementControlVisibility("medical\_history\_specify", true);

}

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("medical\_allergies"))) {

\_surveyPanel.setSurveyElementControlVisibility("medical\_allergies\_specify", true);

}

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("similar\_skin\_condition"))) {

\_surveyPanel.setSurveyElementControlVisibility("skin\_condition\_recommendation", true);

}

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("chemical\_exposure"))) {

\_surveyPanel.setSurveyElementControlVisibility("exposure\_recommendation", true);

}

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("adverse\_drug\_reaction"))) {

\_surveyPanel.setSurveyElementControlVisibility("adverse\_recommendation", true);

}

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("photosensitizing\_drug"))) {

\_surveyPanel.setSurveyElementControlVisibility("photosentisizing\_recommendation", true);

}

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("diagnosed\_headache"))) {

\_surveyPanel.setSurveyElementControlVisibility("pharmacologic\_what", true);

\_surveyPanel.setSurveyElementControlVisibility("pharmacologic\_effect", true);

}

// This is to activate the apointment on submittal and dynamically set the appointment date based on the follow up date

if (!((\_surveyPanel.getResponseForElementId("follow\_up\_date")) == null || "".equalsIgnoreCase(\_surveyPanel.getResponseForElementId("follow\_up\_date")))) {

\_directiveMap.put("SUBMIT\_REGISTRATION","CREATE\_APPOINTMENT=YES");

DateTimeFormatter fmt = DateTimeFormat.forPattern("yyyy-MM-dd");

DateTime fDate = fmt.parseDateTime(\_surveyPanel.getResponseForElementId("follow\_up\_date"));

DateTime cDate = new DateTime();

Days diff = Days.daysBetween(cDate.toLocalDate(),fDate.toLocalDate());

String app\_days = diff.getDays().toString();

\_logger.debug("Survey - 80357 Atopic Dermatitis Initial >> " + app\_days + " days till follow up");

\_directiveMap.put(AgentDirective.APPOINTMENT\_STARTDATE\_DAYS, app\_days);

String msg = "Call patient to follow-up on their atopic dermatitis assessment by ";

String pharmacistName = "";

com.pasi.agentdb.service.UserService userService = \_agentConfig.getUserService();

String pharmacistId = \_surveyPanel.getResponseForElementId("ph\_id");

boolean isPharmacistOidNumeric = true;

if(pharmacistId != null && !pharmacistId.equals("")) {

for(char c: pharmacistId.toCharArray()) {

if(!Character.isDigit(c)) {

isPharmacistOidNumeric = false;

break;

}

}

}

if(!isPharmacistOidNumeric) {

int firstSegment = pharmacistId.indexOf('^');

pharmacistId = pharmacistId.substring(0, firstSegment);

}

if(pharmacistId != null && !pharmacistId.equals("")) {

com.pasi.agent.User pharmacist = userService.loadUser(new Long(pharmacistId));

if(pharmacist != null) {

pharmacistName = pharmacist.getFirstname() + " " + pharmacist.getLastname();

}

}

if(pharmacistName == null ) pharmacistName = "" ;

msg = msg + pharmacistName + " on " + fmt.print(cDate) + ".";

\_logger.debug(msg);

\_directiveMap.put(AgentDirective.APPOINTMENT\_DESCRIPTION,msg);

}

if ("No".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("similar\_skin\_condition"))

&& "No".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("chemical\_exposure"))

&& "No".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("adverse\_drug\_reaction"))

&& "No".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("photosensitizing\_drug"))

) {

\_surveyPanel.setSurveyElementControlVisibility("no\_response\_logic", true);

\_surveyPanel.setSurveyElementControlVisibility("label5", true);

\_surveyPanel.setSurveyElementControlVisibility("consistent\_atopic\_dermatitis", true);

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("consistent\_atopic\_dermatitis"))) {

\_surveyPanel.setSurveyElementControlVisibility("symptoms\_dermatitis", true);

if (( \_surveyPanel.getResponseForElementId("symptoms\_dermatitis")).indexOf("5") > -1) {

\_surveyPanel.setSurveyElementControlVisibility("symptoms\_dermatitis\_note", true);

}

}

\_surveyPanel.setSurveyElementControlVisibility("lesions\_location", true);

\_surveyPanel.setSurveyElementControlVisibility("large\_skin\_involved", true);

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("large\_skin\_involved"))) {

\_surveyPanel.setSurveyElementControlVisibility("large\_skin\_recommendation", true);

}

\_surveyPanel.setSurveyElementControlVisibility("lesions\_face", true);

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("lesions\_face"))) {

\_surveyPanel.setSurveyElementControlVisibility("moderate\_recommendation", true);

}

\_surveyPanel.setSurveyElementControlVisibility("infection\_symptoms", true);

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("infection\_symptoms"))) {

\_surveyPanel.setSurveyElementControlVisibility("infection\_recommendation", true);

}

if (!(( \_surveyPanel.getResponseForElementId("symptoms\_dermatitis")).indexOf("5") > -1 && "Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("consistent\_atopic\_dermatitis")))

&& "No".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("large\_skin\_involved"))

&& "No".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("lesions\_face"))

&& "No".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("infection\_symptoms"))

) {

\_surveyPanel.setSurveyElementControlVisibility("no\_symptoms\_logic", true);

\_surveyPanel.setSurveyElementControlVisibility("label\_treatment", true);

\_surveyPanel.setSurveyElementControlVisibility("recommended\_treatment", true);

if (!"".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("recommended\_treatment"))) {

if (( \_surveyPanel.getResponseForElementId("recommended\_treatment")).contains("3")){

\_surveyPanel.setSurveyElementControlVisibility("infant\_recommendation", true);

\_surveyPanel.setSurveyElementControlVisibility("older\_child\_recommendation", true);

\_surveyPanel.setSurveyElementControlVisibility("older\_child\_recommendation2", true);

}

\_surveyPanel.setSurveyElementControlVisibility("label6", true);

\_surveyPanel.setSurveyElementControlVisibility("issued\_prescription", true);

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("issued\_prescription"))) {

\_surveyPanel.setSurveyElementControlVisibility("rationale\_prescribing", true);

\_surveyPanel.setSurveyElementControlVisibility("medSask\_text", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_textsearch", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_search", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_drug", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_strength", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_din", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_sig", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_qty", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_orig\_fill", true);

\_surveyPanel.setSurveyElementControlVisibility("text\_refills", true);

\_surveyPanel.setSurveyElementControlVisibility("gs\_adapt\_rx\_date", true);

\_surveyPanel.setSurveyElementControlVisibility("text\_footer14", true);

\_surveyPanel.setSurveyElementControlVisibility("counseling", true);

Below is an example configuration of the search button. It is used to access the health Canada database api in order to look up a drug. This code can be reused on other minor ailments if needed.

if( "gs\_search".equals( \_sourceElementId )) {

JComponent button = \_surveyPanel.getControlForElement("gs\_search");

HCDPDApi hcdpdApi = \_agentConfig.getHcdpdApi();

DrugProduct drugProduct = hcdpdApi.drugProductSearchSingleSelect(button);

if(drugProduct != null) {

//set fields

\_surveyPanel.setSurveyElementControlValue("gs\_adapt\_din",drugProduct.din);

//\_surveyPanel.setSurveyElementControlValue("gs\_adapt\_drug",drugProduct.getBrandName()); //option for product name only

\_surveyPanel.setSurveyElementControlValue("gs\_adapt\_drug",drugProduct.drugNameWithIngredients(150)); //option for name and ingredient

\_surveyPanel.setSurveyElementControlValue("gs\_adapt\_strength",drugProduct.getStrength());

}

}

}

}

}

if ((( \_surveyPanel.getResponseForElementId("symptoms\_dermatitis")).indexOf("5") > -1 && "Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("consistent\_atopic\_dermatitis")))

|| "Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("large\_skin\_involved"))

|| "Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("lesions\_face"))

|| "Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("infection\_symptoms"))

) {

\_surveyPanel.setSurveyElementControlVisibility("yes\_symptoms\_logic", true);

}

}

if ("Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("similar\_skin\_condition"))

|| "Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("chemical\_exposure"))

|| "Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("adverse\_drug\_reaction"))

|| "Yes".equalsIgnoreCase( \_surveyPanel.getResponseForElementId("photosensitizing\_drug"))

) {

\_surveyPanel.setSurveyElementControlVisibility("yes\_response\_logic", true);

}

} catch ( Throwable t ) {

\_logger.error("Error evaluating Expression " + t.getClass().getName(), t );

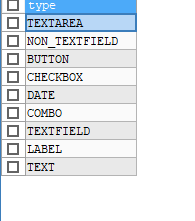
\_surveyPanel.setSurveyElementControlValue("gs\_province", "Expression Error");

}

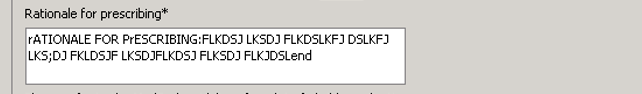
**Survey Element**

The survey element table is where the actual look and feel part of the survey panel will be configured. The order that the elements will appear in is controlled by the sequence column. There cannot be any gaps in the sequence if the survey is to function correctly.

Minior ailments uses the following type of elements:



TEXT AREA: Text areas look like this:



The db entry for the above would look like this: (minus the example text. I have not included the auto generated oid number):

insert into `survey\_element` (`id`, `type`, `prompt`, `default\_value`, `choice\_list`, `mandatory`, `element\_layout`, `control\_layout`, `max\_response\_length`, `survey\_oid`, `sequence`, `subpanel\_id`, `subpanel\_sequence`, `prompt\_fr`, `default\_value\_fr`, `choice\_list\_fr`, `inclusion\_filter\_script`, `mandatory\_expression`, `edit\_mask`, `properties`, `visiblity\_expression`, `create\_listener`, `text\_columns`, `text\_rows`, `group\_id`, `disabled`) values('rationale\_prescribing','TEXTAREA','Rationale for prescribing\*',NULL,NULL,'0','VERTICAL','VERTICAL','100','80357','44',NULL,NULL,'Rationale for prescribing\*',NULL,NULL,NULL,'<BEANSHELL\_EXPRESSION>((\"Yes\".equalsIgnoreCase(\_survey.getElementValue(\"issued\_prescription\"))))',NULL,NULL,NULL,NULL,NULL,NULL,NULL,'0');

Here is an explanation of each field. Unless otherwise noted this will be the same functionality for other element types:

ID: unique per survey identifier for the element used not only on the survey table but can be used on some other supporting tables or groovy scripts

Type: The type as per list above.

Prompt : This is the text that will appear on the screen before an element. Usually posed as a question.

Default\_value: Used to control a prepopulated value in that field

Choice\_list: Used on some element types such as combo to define the options

Mandatory: Defines if a question must be answered or not. This can be overridden on another field however

Element layout: Changes look and feel of element if horizontal or vertical are used

Max\_response\_length: Used to limit the amount of data in text fields. Note that this also influences the size of the element.

Survey oid: Used to reference the survey to elements

Sequence: order in which elements appear

subpanel\_id: Not used in minor ailments.

subpanel\_sequence: Not used in minor ailments.

prompt\_fr: Same as above but French

default\_value\_fr: Same as above but French

choice\_list\_fr: Same as above but French \*\*\*\* SPECIAL \*\*\*\*: Note The number of elements in the English and French choice list must be the same or an error will occur

inclusion\_filter\_script: Not used in minor ailments

mandatory\_expression: Used as an override to determine if something is mandatory. Usually based on the response in another element: Example

<BEANSHELL\_EXPRESSION>(("Yes".equalsIgnoreCase(\_survey.getElementValue("diagnosed\_headache"))))

edit\_mask: Not used in minor ailments

properties Not used in minor ailments

visiblity\_expression Not used in minor ailments

create\_listener Not used in minor ailments

text\_columns Not used in minor ailments

text\_rows Not used in minor ailments

group\_id Not used in minor ailments

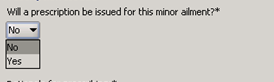
disabled Not used in minor ailments

to continue the other elements types look like this:

Non\_Textfield:



Combo:



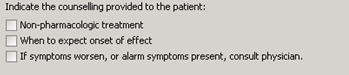
Label:



Button:



Checkbox:



Date:



Text field:



Text:



Note field text supports the HTML tag to do advanced formatting.

Example:

<html><b>Older child, adult:</b> face, skin folds = prescribe low potency corticosteroid

**Program DIN**

Determines which stores have a program. The values are used in the following way:

`oid` - Auto generated

`pharmacy\_oid` - The oid of the pharmacy that will have the program

`exclude\_pharmacy` - Always set to 0 for minor ailments

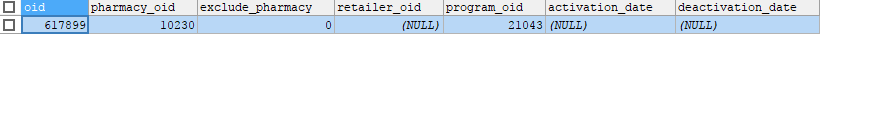
`retailer\_oid`, - Always set to null for minor ailments

`program\_oid`, - The oid of the minor ailments program

`activation\_date`, - Currently unused. Set to null

`deactivation\_date`- Currently unused. Set to null

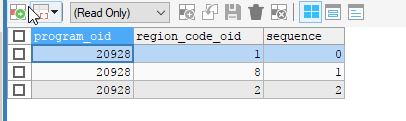
Example entry:



**Program region codes**

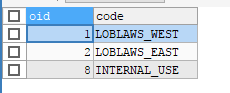
Determines which regions are allowed to get a program. It is mainly used as an additional safety check to not allow programs into an unwanted region/retailer however it can be used for programs that are activated.

Example program oid 20928



This program oid active on 3 regions. The reference for which regions those are is part of the region code table:

Here are the matching entries



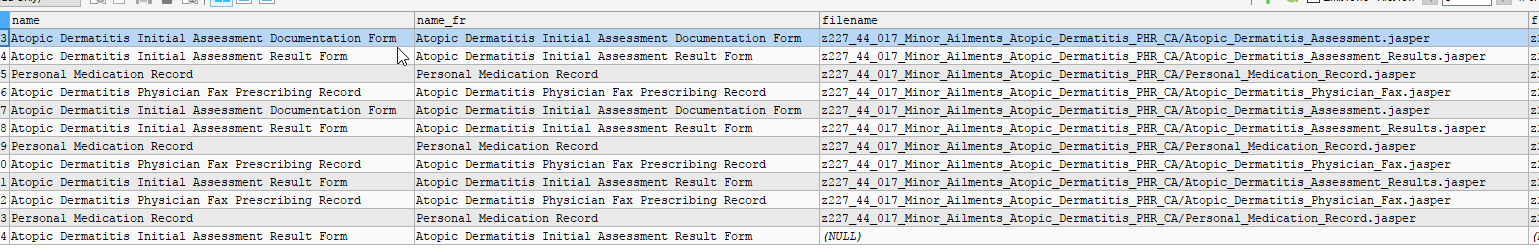
**Program filter conditions**

Not currently used in minor ailments

**Program directive report**

This table controls the documents that are available to be printed as well as other actions that happen be it on launch or on completion. For example for program oid 21043

We have entries like this:

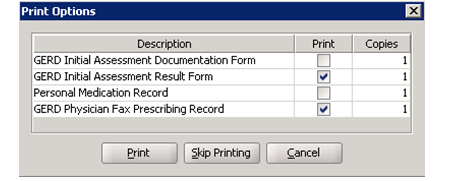


Example entry:

insert into `program\_directive\_report` (`name`, `name\_fr`, `filename`, `filename\_fr`, `graphics\_filenames`, `binding`, `default\_copies`, `track\_print`, `prompt\_user`, `event\_registration\_type`, `program\_oid`, `sequence`, `print\_type`, `report\_type`, `report\_id`, `merge\_report\_ids`, `required\_datasource`, `print\_flag\_expression`, `fax\_flag`, `fax\_number`, `fax\_phone\_number`, `fax\_recipient`, `filter\_expression`) values('Atopic Dermatitis Initial Assessment Documentation Form','Atopic Dermatitis Initial Assessment Documentation Form','z227\_44\_017\_Minor\_Ailments\_Atopic\_Dermatitis\_PHR\_CA/Atopic\_Dermatitis\_Assessment.jasper','z227\_44\_017\_Minor\_Ailments\_Atopic\_Dermatitis\_PHR\_CA/Atopic\_Dermatitis\_Assessment.jasper','z227\_44\_017\_Minor\_Ailments\_Atopic\_Dermatitis\_PHR\_CA/AtopicDermatitis\_Initial.groovy,z227\_44\_017\_Minor\_Ailments\_Atopic\_Dermatitis\_PHR\_CA/arrow.jpg','PRINT\_FORM','0','0','1','MARKET\_SURVEY\_PRINT','21043','0','PRINTER\_PROMPT',NULL,NULL,NULL,NULL,'<AGENT\_BEANSHELL\_EXPRESSION>\r\n{\r\nreturn true;\r\n}\r\n','0',NULL,NULL,NULL,NULL);

You will notice that some of the forms have multiple entries. This is because each entry for a form has a different binding. This binding controls the action the form is used on but first I’ll go over the rest of the fields.

Name – Name of how document will appear if presented to user in print options: Example print screen



name\_fr: Same as English field but French language

`filename`: Path and file name used. Will reference a .Jasper file. NOTE\*\* If updating a file that is already in production a version number will need to be appended to the file name to prevent a cached file from being used. Example file1.jasper needs to be come file1\_v2.jasper

, `filename\_fr` Same as English field but French language

, `graphics\_filenames` - Path and file names of files used to support the jasper. This may include sub reports, groovy scripts as well as graphics. Example:

z227\_44\_017\_Minor\_Ailments\_Atopic\_Dermatitis\_PHR\_CA/AtopicDermatitis\_Initial.groovy,z227\_44\_017\_Minor\_Ailments\_Atopic\_Dermatitis\_PHR\_CA/arrow.jpg

`binding` - Minor ailments uses 3 bindings:

PRINT\_FORM – Used for when print is selected normally on main panel.

PRINT\_BEFORE\_SUBMIT\_REGISTRATION – Used for when user is given option to print when they hit complete.

SUBMIT\_REGISTRATION – Used to force print document or sent document to patient file after registration complete.

, `default\_copies`- Controls default number of copies

, `track\_print` - Flag used if we want to report on the number of prints of the document

, `prompt\_user`, - Flag used to make the print prompt the user or not

`event\_registration\_type` - Controls type of registration. The value of “MARKET\_SURVEY\_PRINT” should be used here.

, `program\_oid` - The program oid of the program

, `sequence` - Controls order of how options are presented to user

, `print\_type` - 3 different types used here for minor ailments:

PRINTER\_PROMPT – Prompts the user to print

MERGE\_FILE – Used to combine the prints of several documents into 1 pdf

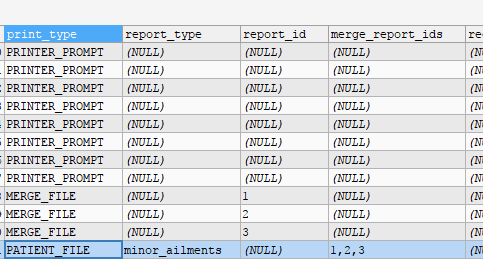
PATIENT\_FILE – Used to send a copy of the document to patient file. Often used with merge file to create 1 document on patient file for a registration.

, `report\_type` - used to define merge report

, `report\_id` ` - used to define merge report

, `merge\_report\_ids` ` - used to define merge report

For the 3 fields above here is an example of how they are used:



, `required\_datasource` - not used in minor ailments

, `print\_flag\_expression`- Controls if a document will print: Example

<AGENT\_BEANSHELL\_EXPRESSION>

import com.pasi.survey.AgentSurveyScriptHelper;

import org.apache.commons.lang3.StringUtils;

import com.pasi.Survey;

Survey survey = \_directiveMap.get(com.pasi.common.SimpleSurveyDirective.PROGRAM\_SURVEY);

String response = Survey.retrieveSurveyResponseForElementId( survey.getQuestionResponseMap(), "issued\_prescription");

return ( response.equalsIgnoreCase("Yes") );

IN the above example a survey element is referenced by id.

, `fax\_flag` not used in minor ailments

, `fax\_number` not used in minor ailments

, `fax\_phone\_number` not used in minor ailments

, `fax\_recipient` not used in minor ailments

, `filter\_expression` used in conjuction with print flag expression: Example

<AGENT\_BEANSHELL\_EXPRESSION>

import com.pasi.survey.AgentSurveyScriptHelper;

import org.apache.commons.lang3.StringUtils;

import com.pasi.Survey;

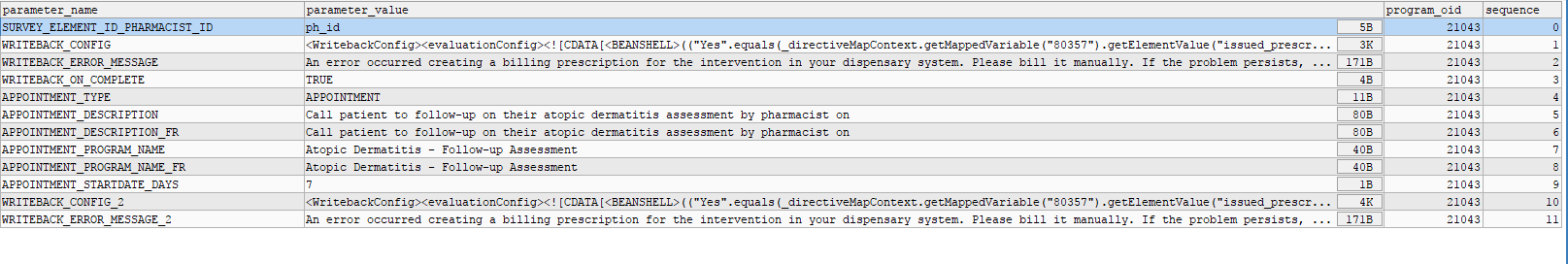
Survey survey = \_directiveMap.get(com.pasi.common.SimpleSurveyDirective.PROGRAM\_SURVEY);

String response = Survey.retrieveSurveyResponseForElementId( survey.getQuestionResponseMap(), "issued\_prescription");

return ( response.equalsIgnoreCase("Yes") );

**Program parameter**

Used to configure other parameters for the program:



\*\* Note some additional explanation will need to be added.

Sample output form:

As you can see the form below maps the majority of fields to a question within the survey panel. The form below is a .jasper file that has been created through ireport.

