Table of Contents

[Screen Shot of EXTSET01 2](#_Toc323745491)

[Overview 2](#_Toc323745492)

[Forms Startup 2](#_Toc323745493)

[Main Menu 3](#_Toc323745494)

[Exit Button 3](#_Toc323745495)

[Setup Criteria (#1) 3](#_Toc323745496)

[Submit Button (#2) 3](#_Toc323745497)

[Submit Button 3](#_Toc323745498)

[Progress (#3) 4](#_Toc323745499)

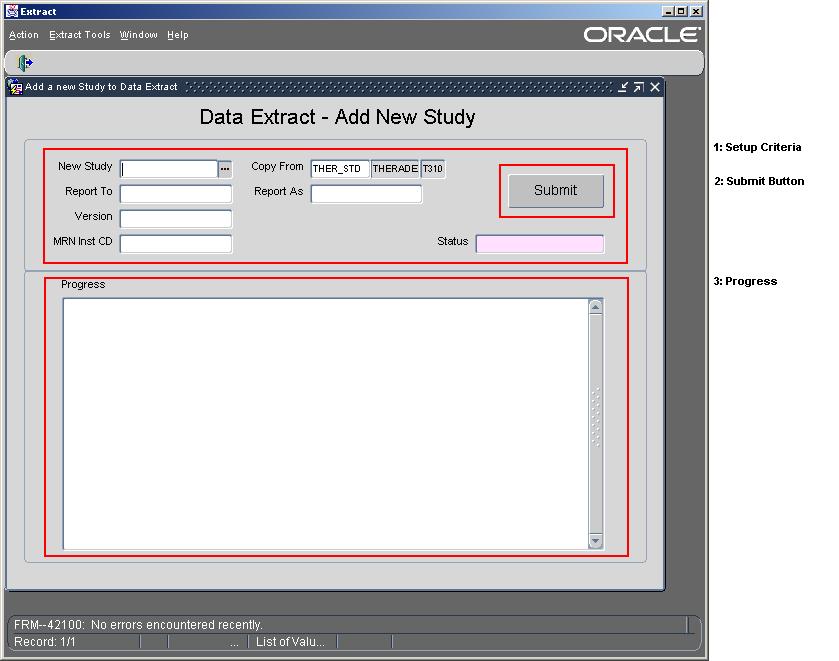
[TRIGGER CODE 5](#_Toc323745500)

[SUBMIT BUTTON 5](#_Toc323745501)

Detailed analysis of EXTSET01.FMB

Note: Used TEXTPAD to look inside form as the 6i Form Builder was unable to read form.

# Screen Shot of EXTSET01



# Overview

This form is used to create new definitions for extraction. It uses a previously created study or template study definition to create the new definition.

# Forms Startup

1. Security Checked at form startup. The following query checks userid to see if user is valid for APPLICATION. (WHEN-NEW-FORM-INSTANCE)

SELECT '1' INTO T\_WHAT

FROM CT\_EXT\_ACCOUNTS

WHERE (USER\_NAME = USER or User = 'CTEXT\_T')

AND ROWNUM = 1;

1. Menu Actions are turn off that would normally allow queries and editing.

# Main Menu

The main menu controls what other screens can be selected. They are controlled through ROLES. Only certain Oracle ROLES granted to the user would allow access. See EXTMENU1\_Analysis for more information.

# Exit Button

The exit button is used to close the application. When closing the application, the screen is checked for existing non-committed transactions and will not quite until the transactions are committed or rolled back.

# Setup Criteria (#1)

There are 6 enterable fields that make up the seteup criteria, New Study, Copy From, Report To, Report As, Version and MRN Inst CD

1. New Study: The list of available studies, matched against already defined studies. The “Yes” indicator denotes ta study having a definition:

FROM ocl\_studies a,

(SELECT OC\_STUDY, 'Yes' a\_ind from CT\_EXT\_STUDY\_RPT\_CTL group by OC\_STUDY) b

where a.study = b.oc\_study (+)

1. Copy From: This is actually 3 fields consisting of Study, Report To and Version. The LOV for this field shows a list of already created definitions. Making a selection from the list fills in all three fields. Making a selection from the list will also file in the Report To and Version Set Up Fields mentioned below. It appears to only do this when they are blank. Query for LOV:

FROM CT\_EXT\_STUDY\_RPT\_CTL GROUP BY OC\_STUDY, REPORT\_TO, VERSION ORDER BY

DECODE(OC\_STUDY, 'THER\_STD', '0', OC\_STUDY)

NOTE: The DECODE make it so the THER\_STD study is at the TOP of the list.

NOTE2: Oracle LOV have self reducing ability…need to see if this is available in HTML

1. Report To: Text field only, Not validated (Should it?)
2. Version: Text field only, Not validated (Should it?)
3. Report As: Text field only, Not validated (Should it?)
4. Version: Text field only, Not validated (Should it?)
5. MRN Inst CD: Text field only, Not validated (Should it?)
6. Status: Output Field. Updated with running/final status of execution of the “Submit” button.

# Submit Button (#2)

#### Submit Button

The Submit Button is used to create a study extraction definition as a copy of another study definition. It is the primary function of this application.

1. Only fires if OC\_STUDY, REPORT\_TO, VERSION, MRN\_INST\_CD and COPY\_OC\_STUDY fields have been entered.

NOTE: Report AS should also NOT be blank.

1. Checks to see if Target Study / Report To is the same as “Copy From” Study / Report To and stops if they are the same.
2. Checks to see if Target Study / Report To already exists in control definition table and only continues if user states overwrite is okay.

Note: May have to make this a Check Box button.

1. Checks to see if “Copy From” Study / Report To exists in definition table CT\_EXT\_STUDY\_RPT\_CTL. If it does not exists, stops with warning message

NOTE: It seems that these queries should be checking for Study/Report To/Version instead of Study/Report To. This needs verified.

1. Processing Starts:

Generates Object Owner and Extract Owner Names

1. Checks for existence of SNAP Shot Views, stops execution if they do not exist.

EXT\_STUP\_PKG.COUNT\_SNAPSHOT\_VWS(T\_OC\_OBJECT\_OWNER);

1. Creates Definition while updating progress Screen (See Full Cursor Code for Details)

NOTE: This should be closely examined as the Web App may not be able to transaction each of these steps. Most of the steps are DDL statements and auto commit, so it might not matter.

# Progress field (#3)

The Progress field contains a running commentary of the steps executed and automatically refreshes. As each step requires a previous step, this will need to be scrutinized closely.

# TRIGGER CODE

#### SUBMIT BUTTON

/\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*/

/\* Modification History \*/

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

/\* PRC - Ekagra 2/19/2009 \*/

/\* PRIMARY OBJECTIVE: \*/

/\* Altered Trigger and removed all statements that would require system level \*/

/\* database privileges and placed those statements into the package EXT\_STUP\_PKG. \*/

/\* By doing this, individual users having roles that have system privileges \*/

/\* will be can be removed. The OWNER of the package will instead have the \*/

/\* the privileges granted and the users will be granted EXECUTE privilege to the \*/

/\* new package. \*/

/\* SECONDARY OBJECTIVE: \*/

/\* Cleaned up the code. \*/

/\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*/

T\_OC\_STUDY VARCHAR2(30);

T\_REPORT\_TO VARCHAR2(30);

T\_REPORT\_AS VARCHAR2(30);

T\_VERSION VARCHAR2(30);

T\_MRN\_INST\_CD VARCHAR2(30);

T\_COPY\_OC\_STUDY VARCHAR2(30);

T\_COPY\_REPORT\_TO VARCHAR2(30);

T\_COPY\_VERSION VARCHAR2(30);

T\_SQL\_STR VARCHAR2(2000);

T\_ACTION VARCHAR2(2000);

T\_OC\_STUDY\_ID NUMBER;

T\_COPY\_OC\_STUDY\_ID NUMBER;

T\_OC\_OBJECT\_OWNER VARCHAR2(30);

T\_EXT\_OBJECT\_OWNER VARCHAR2(30);

T\_TEMP NUMBER;

T\_TEXT LONG;

T\_COUNT NUMBER := 0;

crlf varchar2(2) := Chr(10);

alert\_button NUMBER;

T\_ALERT\_MSG VARCHAR2(200);

T\_ALERT\_TITLE VARCHAR2(200);

BEGIN

GO\_FIELD('CG$CTRL.RUN\_PROGRESS');

:CG$CTRL.RUN\_STATUS\_TEXT := 'IN PROGRESS';

:CG$CTRL.RUN\_PROGRESS := NULL;

IF RTRIM(:STUDY\_RPT.OC\_STUDY) IS NULL OR RTRIM(:STUDY\_RPT.REPORT\_TO) IS NULL OR

RTRIM(:STUDY\_RPT.VERSION) IS NULL OR RTRIM(:STUDY\_RPT.MRN\_INST\_CD) IS NULL OR

RTRIM(:CG$CTRL.COPY\_OC\_STUDY) IS NULL THEN

T\_ALERT\_TITLE := 'Warning ';

T\_ALERT\_MSG := 'Please enter the required fields.';

SET\_ALERT\_PROPERTY('ALERT\_STOP\_1', ALERT\_MESSAGE\_TEXT, T\_ALERT\_MSG);

SET\_ALERT\_PROPERTY('ALERT\_STOP\_1', TITLE, T\_ALERT\_TITLE);

alert\_button := Show\_Alert('ALERT\_STOP\_1');

IF alert\_button = ALERT\_BUTTON1 THEN

:CG$CTRL.RUN\_STATUS\_TEXT := 'CANCELLED';

RETURN;

END IF; -- do nothing if operator selects button 2 cancel

RETURN; -- prc: check this

END IF;

IF :STUDY\_RPT.OC\_STUDY||'-'||:STUDY\_RPT.REPORT\_TO = :CG$CTRL.COPY\_OC\_STUDY||'-'||:CG$CTRL.COPY\_REPORT\_TO THEN

T\_ALERT\_TITLE := 'Warning ';

T\_ALERT\_MSG := 'You can not copy from the same study and the same extract.';

SET\_ALERT\_PROPERTY('ALERT\_STOP\_1', ALERT\_MESSAGE\_TEXT, T\_ALERT\_MSG);

SET\_ALERT\_PROPERTY('ALERT\_STOP\_1', TITLE, T\_ALERT\_TITLE);

alert\_button := Show\_Alert('ALERT\_STOP\_1');

IF alert\_button = ALERT\_BUTTON1 THEN

:CG$CTRL.RUN\_STATUS\_TEXT := ' CANCELLED';

RETURN;

END IF; -- do nothing if operator selects button 2 cancel

RETURN;

END IF;

BEGIN

-- Check to see if Extract for Study Exists by checking REPORT Control Table

SELECT 1 INTO T\_TEMP

FROM CT\_EXT\_STUDY\_RPT\_CTL

WHERE OC\_STUDY = :STUDY\_RPT.OC\_STUDY

AND REPORT\_TO = :STUDY\_RPT.REPORT\_TO

AND ROWNUM = 1;

T\_ALERT\_TITLE := 'Warning - Study found in extract. ';

T\_ALERT\_MSG := 'This Study already exists in Extract. Do you want to overwrite?';

SET\_ALERT\_PROPERTY('ALERT\_WARN\_1', ALERT\_MESSAGE\_TEXT, T\_ALERT\_MSG);

SET\_ALERT\_PROPERTY('ALERT\_WARN\_ 1', TITLE, T\_ALERT\_TITLE);

alert\_button := Show\_Alert('ALERT\_WARN\_1');

IF alert\_button = ALERT\_BUTTON2 THEN

:CG$CTRL.RUN\_STATUS\_TEXT := ' CANCELLED';

RETURN;

END IF; -- do nothing if operator selects button 2 cancel

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

NULL;

END;

IF (:CG$CTRL.COPY\_OC\_STUDY IS NOT NULL) THEN

BEGIN

-- Check to see if Extract for TEMPLATE Study Exists in Report Control table

SELECT 1 INTO T\_TEMP

FROM CT\_EXT\_STUDY\_RPT\_CTL

WHERE OC\_STUDY = :CG$CTRL.COPY\_OC\_STUDY

AND REPORT\_TO = :CG$CTRL.COPY\_REPORT\_TO --:STUDY\_RPT.REPORT\_TO

AND ROWNUM = 1;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

T\_ALERT\_TITLE := 'Warning - No Source Study. ';

T\_ALERT\_MSG := 'The Source study does not exist. Please select another one.';

SET\_ALERT\_PROPERTY('ALERT\_STOP\_1', ALERT\_MESSAGE\_TEXT, T\_ALERT\_MSG);

SET\_ALERT\_PROPERTY('ALERT\_STOP\_1', TITLE, T\_ALERT\_TITLE);

alert\_button := Show\_Alert('ALERT\_STOP\_1');

IF alert\_button = ALERT\_BUTTON1 THEN

:CG$CTRL.RUN\_STATUS\_TEXT := ' CANCELLED';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

RETURN;

END IF; -- do nothing if operator selects button 2 cancel

END;

END IF;

-- PRC : Check This : Turn off the SUBMIT Button

IF GET\_ITEM\_PROPERTY('CG$CTRL.BUT\_SUBMIT',ENABLED) = 'TRUE' THEN

SET\_ITEM\_PROPERTY('CG$CTRL.BUT\_SUBMIT',ENABLED,PROPERTY\_FALSE);

END IF;

T\_OC\_STUDY := RTRIM(LTRIM(:STUDY\_RPT.OC\_STUDY));

T\_REPORT\_TO := RTRIM(LTRIM(:STUDY\_RPT.REPORT\_TO));

T\_REPORT\_AS := RTRIM(LTRIM(:STUDY\_RPT.REPORT\_AS));

T\_VERSION := RTRIM(LTRIM(:STUDY\_RPT.VERSION));

T\_MRN\_INST\_CD := RTRIM(LTRIM(:STUDY\_RPT.MRN\_INST\_CD));

T\_COPY\_OC\_STUDY := RTRIM(LTRIM(:CG$CTRL.COPY\_OC\_STUDY));

T\_COPY\_REPORT\_TO := RTRIM(LTRIM(:CG$CTRL.COPY\_REPORT\_TO));

T\_COPY\_VERSION := RTRIM(LTRIM(:CG$CTRL.COPY\_VERSION));

-- Check Study name, must not start with Numeric, study name is used in new Schema Name

IF SUBSTR(T\_OC\_STUDY, 1, 1) IN ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9') THEN

T\_OC\_OBJECT\_OWNER := 'S'||T\_OC\_STUDY||'$SNAPSHOT';

T\_EXT\_OBJECT\_OWNER := 'S'||T\_OC\_STUDY||'\_EXT';

ELSE

T\_OC\_OBJECT\_OWNER := T\_OC\_STUDY||'$SNAPSHOT';

T\_EXT\_OBJECT\_OWNER := T\_OC\_STUDY||'\_EXT';

END IF;

BEGIN

-- Get a count of the Snapshot Views of the Owner

T\_Temp := EXT\_STUP\_PKG.COUNT\_SNAPSHOT\_VWS(T\_OC\_OBJECT\_OWNER);

IF T\_TEMP = 0 THEN

T\_ALERT\_TITLE := 'Warning - No Snap Shot Views. ';

T\_ALERT\_MSG := 'Please create the snapshot views before proceeding.';

SET\_ALERT\_PROPERTY('ALERT\_STOP\_1', ALERT\_MESSAGE\_TEXT, T\_ALERT\_MSG);

SET\_ALERT\_PROPERTY('ALERT\_STOP\_1', TITLE, T\_ALERT\_TITLE);

alert\_button := Show\_Alert('ALERT\_STOP\_1');

IF alert\_button = ALERT\_BUTTON1 THEN

:CG$CTRL.RUN\_STATUS\_TEXT := 'CANCELLED';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

RETURN;

END IF; -- do nothing if operator selects button 2 cancel

/\* MESSAGE('Please create the snapshot views before proceeding.', ACKNOWLEDGE);

IF GET\_ITEM\_PROPERTY('CG$CTRL.BUT\_SUBMIT',ENABLED) = 'FALSE' THEN

SET\_ITEM\_PROPERTY('CG$CTRL.BUT\_SUBMIT',ENABLED,PROPERTY\_TRUE);

END IF;

RETURN; \*/

END IF;

END;

SELECT TASK\_ID INTO T\_OC\_STUDY\_ID

FROM OCL\_STUDIES

WHERE STUDY = T\_OC\_STUDY;

SELECT TASK\_ID INTO T\_COPY\_OC\_STUDY\_ID

FROM OCL\_STUDIES

WHERE STUDY = T\_COPY\_OC\_STUDY;

:CG$CTRL.RUN\_PROGRESS := '"ADD STUDY" BEGINING...';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

-- 1. Set up a user account----

T\_ACTION := 'Setting up extract account ';

BEGIN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||CRLF||'..'||T\_ACTION||'...';

Synchronize; -- P RC 06/13/06 : Updates screen while action takes place.

EXT\_STUP\_PKG.SET\_STUDY\_CTRL(T\_OC\_STUDY, T\_OC\_OBJECT\_OWNER, T\_EXT\_OBJECT\_OWNER);

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'. . . . .Study Control Setup -- Done.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXT\_STUP\_PKG.BUILD\_EXT\_SCHEMA(T\_EXT\_OBJECT\_OWNER);

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||CRLF||'. . . . .Extract Schema Setup -- Done.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||' -- COMPLETE.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXCEPTION

WHEN OTHERS THEN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||' -- FAIL.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

RAISE;

END;

--2. Setting up Study Reporting ----

T\_ACTION := 'Setting up Study Reporting ';

BEGIN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION;

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXT\_STUP\_PKG.SET\_REPORT\_CTRL(T\_OC\_STUDY, T\_REPORT\_TO, T\_VERSION, T\_REPORT\_AS);

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||' -- COMPLETE.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXCEPTION

WHEN OTHERS THEN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||' -- FAIL.';

Synchronize;

RAISE;

END;

--3. Setting up study extract views ----

T\_ACTION := 'Setting up study extract views ';

BEGIN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||'...';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXT\_STUP\_PKG.SET\_VIEW\_CTRL(T\_OC\_STUDY, T\_MRN\_INST\_CD, T\_COPY\_OC\_STUDY);

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'. . . . .Extract Study Views Setup -- Done.';

Synchronize;

EXT\_MISC\_PKG.P\_GEN\_LAB\_OTH\_VW\_TEXT(T\_OC\_STUDY,T\_VERSION); -- prc 06/08/06 added version

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'. . . . .Extract Study Lab Views Setup -- Done.';

Synchronize;

EXT\_MISC\_PKG.P\_CRT\_STUDY\_VW(T\_OC\_STUDY, 'N',T\_VERSION); -- prc 06/08/06 added version

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'. . . . .Extract Study Views Create -- Done.';

Synchronize;

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||' -- COMPLETE.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXCEPTION

WHEN OTHERS THEN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||' -- FAIL.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

RAISE;

END;

--4. Setting up synonyms ----

T\_ACTION := 'Setting up synonyms ';

BEGIN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION;

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXT\_MISC\_PKG.P\_CRT\_STUDY\_OC\_SYN(T\_OC\_STUDY);

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||' -- COMPLETE.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXCEPTION

WHEN OTHERS THEN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||' -- FAIL.';

Synchronize;

RAISE;

END;

--5. Setting up other objects ----

T\_ACTION := 'Setting up other objects ';

BEGIN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||'...';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXT\_STUP\_PKG.SET\_CRS\_CTRL(T\_OC\_STUDY, T\_REPORT\_TO, T\_VERSION, T\_COPY\_OC\_STUDY, T\_COPY\_REPORT\_TO, T\_COPY\_VERSION);

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'. . . . .Extract Study Cursors Setup -- Done.';

Synchronize;

EXT\_STUP\_PKG.SET\_FILE\_CTRL(T\_OC\_STUDY, T\_REPORT\_TO, T\_VERSION, T\_COPY\_OC\_STUDY, T\_COPY\_REPORT\_TO, T\_COPY\_VERSION);

:CG$CTRL.RUN\_PROGRESS : = :CG$CTRL.RUN\_PROGRESS||crlf||'. . . . .Extract Study File Control Setup -- Done.';

Synchronize;

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||' -- COMPLETE.';

Synchronize; -- PRC 06/13/06 : Updates screen while action takes place.

EXCEPTION

WHEN OTHERS THEN

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'..'||T\_ACTION||' -- FAIL.';

Synchronize;

RAISE;

END;

:CG$CTRL.RUN\_STATUS\_TEXT := 'SUCCESS';

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||'"ADD STUDY" SUCCESS!';

Synchronize;

IF GET\_ITEM\_PROPERTY('CG$CTRL.BUT\_SUBMIT',ENABLED) = 'FALSE' THEN

SET\_ITEM\_PROPERTY('CG$CTRL.BUT\_SUBMIT',ENABLED,PROPERTY\_TRUE);

END IF;

EXCEPTION

WHEN OTHERS THEN

:CG$CTRL.RUN\_STATUS\_TEXT := 'FAILURE';

:CG$CTRL.RUN\_PROGRESS := :CG$CTRL.RUN\_PROGRESS||crlf||

'-- FAILURE.'||crlf||

'SQLError: '||crlf||

Substr(SQLERRM, 1, 250);

Synchronize;

IF GET\_ITEM\_PROPERTY('CG$CTRL.BUT\_SUBMIT',ENABLED) = 'FALSE' THEN

SET\_ITEM\_PROPERTY('CG$CTRL.BUT\_SUBMIT',ENABLED,PROPERTY\_TRUE);

END IF;

RAISE;

END;