eRAD RIS

RELEASE ANNOUNCEMENT

Build 4.2025.054

UPDATED OCTOBER 29, 2025

TABLE OF CONTENTS

PUBLICATION HISTORY

Revision	Author	Description
October 28, 2025	Felicia Ellis, Kevin Brooks	▪ Commercial release.
October 29, 2025	Kevin Brooks	▪ Updated with data fix script via Support #37743

SUMMARY

Release Announcement

This release of eRAD RIS 4.2025.054 enhances Guided Workflow for Mammography by introducing dynamic field visibility, a new Guided Workflow Activity Metrics management report, quick access to the Contact Log, and resolves several Guided Workflow related issues, including an issue with the search range not updating during GW scheduling.

Additionally, this release addresses a RIS to OS integration issue where phantom RIS processes persisted after exit.

This release also merges in the content of Rapid Release eRAD RIS 4.2025.053.0.1 (which resolves an issue where laterality was removed from orders submitted through the Referring Portal when scheduling the exam at a later time).

This release includes a data fix script via Support #37743 that affects customers using Guided Workflow for Mammogram Scheduling.

Refer to Deploy Instructions section for details and manual upgrade steps.

New Features

This release introduces the following features and enhancements:

Category	Redmine #	Subject	Description
Guided Workflow-Mammo	37024	Guided Workflow- Dynamic Elements via Workflow Navigation Agent Access Strings Parameter	This enhancement to Guided Workflow utilizes Access Strings and Permissions returned by the GW Navigation Agent to dynamically modify the behavior of fields (typically to hide them) as users navigate through the workflow.
Guided Workflow-Mammo	37557	Guided Workflow – Activity Metrics Report	This enhancement to Guided Workflow Mammo adds a Guided Workflow Activity Metrics management report.
Guided Workflow-Mammo	37558	Add Button to Quick Open Contact Log	This enhancement to Guided Workflow Mammo adds the ability to quickly open the Contact Log from within the workflow.

Sorted by Category and Redmine

Refer to the Feature Details section below for configuration and usage information.

Resolved Issues

This release resolves the following issues:

Category	Redmine #	Subject
Digital Forms	37654	Resolves a Digital Forms issue with deleting hidden answers.
Digital Forms	37674	Resolves a display issue with Save and Finish Buttons.
Guided Workflow-Mammo	37186	Resolves a display issue with Guidance Panel on first application launch.

Category	Redmine #	Subject
Guided Workflow-Mammo	37743	This backfill script repopulates database c_order_items with values missing from previous Guided Workflow scheduling activities.
Guided Workflow-Mammo	37607	Improves error handling for Guided Workflow Agent and DB connections.
Guided Workflow-Mammo	37616	Resolves Guided Workflow Scheduling issue where the Search Range was not updating-appointment slot search.
OS Projects	37716	Resolves Integration issue where RIS Creates Phantom Processes that interfere with RIS to OS integration.

SORTED BY CATEGORY AND REDMINE

New Known Limitations

The following new Known Limitations were identified with this release:

- EXPECTED BEHAVIOR BY DESIGN (37024)
  The `AddOnly` access level is not supported. Access levels can be set to `None`, `ReadOnly`, `Disabled`, or `Full` only.

Refer to the FEATURE DETAILS section below for more information.

FEATURE DETAILS

Guided Workflow- Mammo

Feature 37024 – Dynamic Elements via Workflow Navigation Agent Access Strings Parameter

Summary

This enhancement to Guided Workflow utilizes Access Strings and Permissions returned by the Guided Workflow Navigation Agent to dynamically modify the behavior of fields (typically to hide them) as users navigate through the workflow.

Background

Existing custom access string functionality (`Custom.Visibility.Data.*`) in RIS has been available for some time, but only maps an explicit access level via the `AccessString` lookup to modify control behavior.

Feature Description

With this update, administrators now have the ability to define control behavior dynamically by creating Guided Workflow navigation rules, without having to set a global state for controls (in `AccessString`) or create new User Groups. Instead, administrators can reference the control in plain language when creating navigation rules and specify the access level to be applied at runtime.

This functionality is only available when using Guided Workflow.

When Guided Workflow users navigate between screens, rules defined in the Workflow Navigation Agent allow parameters to be returned that determine control behavior for each field on the screen to be displayed. When a user navigates to a panel, the system evaluates the `accessStringOverride` property and applies the specified access levels.

This behavior overrides any user or group access string setting that may have been configured in RIS.

Administrators can list any control(s) available on any screen in the workflow (`Patient`, `Order`, `Schedule`, etc.) via custom access strings when typing out their navigation rules.

The `accessStringOverride` property supports the following values:

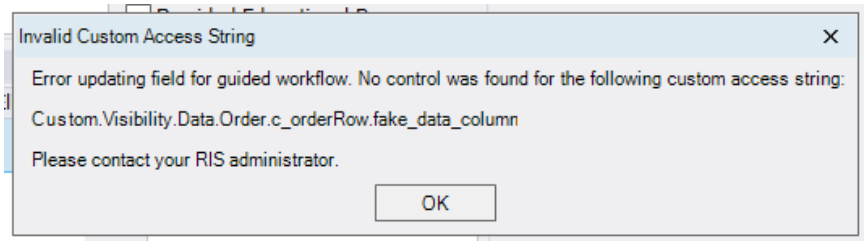
- `accessStringsWithAccessLevelNone`
- `accessStringsWithAccessLevelReadOnly`
- `accessStringsWithAccessLevelDisabled`
- `accessStringsWithAccessLevelFull`

Administrators do not need to reference those properties named above in their rule instructions. Instead, they can write natural language instructions when creating workflow navigation rules using an existing custom access string to define how it should behave, e.g.:

- *When navigating to the Patient screen we need to hide the following fields:*
  - *Custom.Visibility.Data.Patient.c_patientRow.primary_address_line2*
  - *Custom.Visibility.Data.Patient.c_patientRow.alternate_first_name*
  - *Custom.Visibility.Data.Patient.c_patientRow.alternate_last_name*

The system translates this instruction into a structured JSON payload and applies it when the panel loads. This behavior overrides any user or group access string setting that may have been configured in RIS.

If an invalid custom access string is entered, the system checks whether the control exists and displays a warning message if it cannot be found.



Known Limitations

The following significant limitations have been identified and should be communicated to affected users:

- EXPECTED BEHAVIOR BY DESIGN
  - The `AddOnly` access level is not supported. Access levels can be set to `None`, `ReadOnly`, `Disabled`, or `Full` only.

Configuration Instructions

No System Administrator actions are necessary to enable this feature, although

Feature #37557 – Guided Workflow – Activity Metrics Report

Summary

This enhancement to Guided Workflow Mammo adds two new management reports that provide visibility into scheduling and activity metrics within the Guided Workflow.

Background

Previously, Guided Workflow did not include a centralized way to view or track activity metrics across the workflow, requiring the use of SQL queries to retrieve details.

Feature Description

This enhancement introduces two new Guided Workflow reports that consolidate Guided Workflow scheduling data:

- DAILY GUIDED WORKFLOW MAMMOGRAPHY STATISTICS REPORT
- ALL-TIME GUIDED WORKFLOW MAMMOGRAPHY STATISTICS REPORT

The reports summarize key operational metrics such as daily scheduling activity, procedure outcomes, and user performance into a single view for management and analysis.

The reports, titled Guided Workflow for Mammography Statistics and Guided Workflow For Mammography Statistics—All Time, are located under: **Management Reports → Statistics → Schedulers**.

Configuration Instructions

No System Administrator actions are necessary to enable this feature. However, report access may depend on user permissions.

Feature 37558 – Add Button to Quick Open Contact Log

Summary

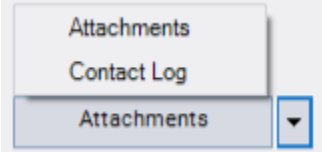
This enhancement to Guided Workflow Mammo adds a new option to quickly open the Contact Log directly from within the workflow.

Background

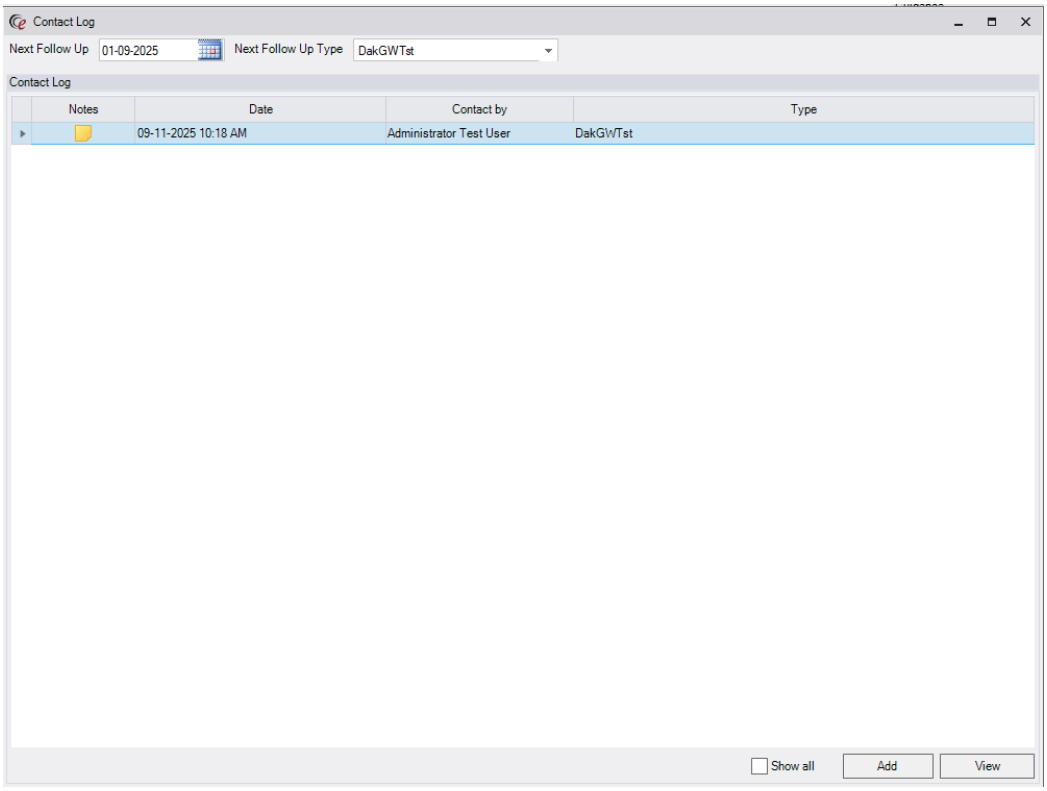
Previously, accessing the Contact Log during Guided Workflow required navigating away from the workflow screen, creating additional steps and interrupting the scheduling process. This change allows users to keep the Contact Log accessible throughout the workflow for greater efficiency and ease of use.

Feature Description

This enhancement updates the Attachments button to a split button with two options: Attachments (default) and Contact Log.



Selecting Contact Log opens the Contact Log in a floating window, allowing the user to continue interacting with the workflow without interruption.



The Contact Log window remains open throughout the workflow and automatically closes when the workflow is completed or exited. If the user attempts to open it again while it's already open, focus will be returned to the existing window.

Configuration Instructions

No System Administrator actions are necessary to enable this feature.

However, if `FollowUp` types are not configured in the system, the Contact Log window will not display the `Show All` checkbox. To ensure the full functionality of the Contact Log within Guided Workflow, Follow up types should be configured in system settings.

DEPLOY INSTRUCTIONS

Standard full package deploy, plus additional step for affected sites that requires Service Team assistance:

Manual Upgrade Steps

Support #37743 - GW - Backfill script for c_order_items

Summary

A high-severity issue has been identified by the Development Team. Exams scheduled via the Guided Workflow for Mammogram Scheduling workflow will have missing c_order_item rows in the database.

A software fix and data fix strategy are available.

AFFECTED VERSION(S)	4.2025.043 THROUGH 4.2025.053
AFFECTED CUSTOMERS	ONLY CUSTOMERS USING GUIDED WORKFLOW FOR MAMMOGRAM SCHEDULING
SEVERITY (LOW/MEDIUM/HIGH/URGENT)	HIGH
IMPACT	DATA INTEGRITY ISSUE

Issue

Exams scheduled via the Guided Workflow for Mammogram Scheduling workflow will have missing c_order_item rows in the database.

This issue was identified by Development. The bug has existed in RIS since the introduction of the Guided Workflow feature in release v4.2025.043.

Impact

When this issue occurs, Referring Portal users accessing any data from affected studies will experience errors.

Workaround

No comprehensive workaround is available without an upgrade.

Expected Action: Upgrade and Data Fix

A resolution to this issue is available with the release of eRAD RIS 4.2025.054 and related Support #37743 - GW - Backfill script for c_order_items.

Perform the following steps to repopulate the missing values for previously impacted records:

1. Recycle portal app pools prior to executing this script to clean up the studies in case some of them have changed in the interim.
2. Execute the following SQL script:

```
001 begin transaction
002
003 -- find study items that where added by our agent and still don't have a c_order_item row assocated.
004 select si.study_item_key, s.order_key, si.procedure_code, si.body_part_code, si.laterality_code, s.patient_key,
    s.study_key into #tmp
005 from
```

```
    c_study_item si inner join
    c_study s on si.study_key = s.study_key
    left outer join c_order_item_x_study_item x on x.study_item_key = si.study_item_key
where
    x.order_item_key is null
    -- reduce the scope as much as possible, only fix what we know we absolutly have to.
    and agent_recommended_flag = 'Y'  and s.status_code <> 'Cancelled' and si.active_flag = 'Y'
    -- eliminate rows that happen to now have a c_order_item for the procedure because those have been handled (somewhat)???
    and not exists (select 1 from c_order osub inner join c_order_item oi on osub.order_key = oi.order_key where osub.order_key = s.order_key and oi.procedure_code = si.procedure_code)
    and si.procedure_code is not null
order by s.study_key desc

declare @study_item_key int
declare @order_key int
declare @patient_key int
declare @study_key int
declare @procedure_code varchar(20)
declare @body_part_code  varchar(20)
declare @laterality varchar(20)

create table #auditKeys(audit_key bigint)

DECLARE recordsToFix CURSOR FOR
select * from #tmp
OPEN recordsToFix;

FETCH NEXT FROM recordsToFix INTO @study_item_key, @order_key, @procedure_code, @body_part_code, @laterality, @patient_key, @study_key
WHILE @@FETCH_STATUS = 0
BEGIN
    insert into c_order_item (order_key, agent_recommended_flag, procedure_code, body_part_code, laterality_code, last_updated, last_updated_by_user_id, sequence_id)
    values(@order_key, 'Y', @procedure_code, @body_part_code, @laterality, SYSDATETIMEOFFSET(), 'redmine_37743',
            -- calculate next sequence_id.
            (select coalesce (max(sequence_id), -1) + 1 from c_order_item oi where oi.order_key = @order_key))

-- then for each of these rows lookup the order_item_key and create the c_order_item_x_study_item row.
    insert into c_order_item_x_study_item(order_item_key, study_item_key, last_updated, last_updated_by_user_id)
    values(@@IDENTITY, @study_item_key, SYSDATETIMEOFFSET(), 'redmine_37743')

    insert into c_audit (description, notes, patient_key, order_key, study_key, last_updated, last_updated_by_user_id,  is_retrieval_flag, procedure_code)
    values(
```

```
    'OrderItem Insert (by technical support)',
    'Redmine Support 37743 for study_item_key: ' + cast(@study_item_key as varchar(20)) ,
    @patient_key,
    @order_key,
    @study_key,
    SYSDATETIMEOFFSET(),
    'RIS Support',
    'N',
    @procedure_code)

  insert into #auditKeys(audit_key) values(@@IDENTITY)

  FETCH NEXT FROM recordsToFix INTO @study_item_key, @order_key, @procedure_code, @body_part_code, @laterality, @patient_key, @study_key
END;

CLOSE recordsToFix;
DEALLOCATE recordsToFix;

-- verify that we got them all.
declare @count int = 0;
select @count = count(*) -- this should be
from
  c_study_item si inner join
  c_study s on si.study_key = s.study_key
  left outer join c_order_item_x_study_item x on x.study_item_key = si.study_item_key
where
  x.order_item_key is null
  -- reduce the scope as much as possible, only fix what we know we absolutly have to.
  and agent_recommended_flag = 'Y'  and s.status_code <> 'Cancelled' and si.active_flag = 'Y'
  -- eliminate rows that happen to now have a c_order_item for the procedure because those have been handled (somewhat)???
  and not exists (select 1 from c_order osub inner join c_order_item oi on osub.order_key = oi.order_key where osub.order_key = s.order_key and oi.procedure_code = si.procedure_code)
  and si.procedure_code is not null

select case when @count = 0 then 'All good' else 'there was a problem, contact dev support before running this in a commit' end
select * from c_audit a inner join #auditKeys aa on a.audit_key = aa.audit_key

rollback transaction
```

VERSION DETAILS

Code Stream

The following source code branches have been merged into this release: