

## Chapter 7: Edit Errors

Computerized edits are automated processes that test the validity of data fields. The following sets of edits are integrated into SEER\*DMS:

- The SEER Edits cover fields submitted to SEER. These edits are also implemented in the SEER\*Edits Windows software.
- The SEER Extended edits were developed by the NCI to extend edits beyond those fields sent to the SEER Program. The extended edits are available to all SEER\*DMS registries and validate fields that are not required to be transmitted to SEER.
- Registry edits are defined and maintained by registry staff. These include edits to validate registry-specific fields, edits to implement validation not covered by SEER edits, and registry-specific versions of SEER edits (the registry may decide to vary the logic of a SEER edit by creating a registry-defined edit and disabling the SEER edit).
- System edits are edits to enforce database constraints. The SEER\*DMS system edits enforce database integrity of fields common to all SEER\*DMS registries. The SEER\*DMS Registry system edits enforce database integrity of registry-specific fields.

SEER, SEER Extended, and all system edits are maintained and deployed by the SEER\*DMS development team. Registry staff are responsible for defining and maintaining registry-specific edits that are not system edits. The SEER\*DMS Edits Manager provides an interface for adding, modifying, and deleting registry edits.

A patient set that is in the workflow will not exit the workflow until all edits are cleared. However, a manual task is not created if an edit error is introduced in a patient set that is not in the workflow. Edit failures caused by modifications made via ad hoc editing or Mass Change imports must be monitored through system reports and corrected in ad hoc editing sessions.

Edits are executed each time a user opens, validates, or saves a record or patient set in SEER\*DMS. Edits are also executed whenever an automated process updates a patient set (this includes updates via Mass Change imports). The *Patient Set Edits System Task* can be used to execute the edits on all patient sets or a cohort of patient sets. The primary purpose of this task is to apply new or modified edits to all patient sets in the database.

In this chapter, you'll learn about

- Understanding Edits in SEER\*DMS
- Viewing Edit Documentation
- Using the Edits Help Page
- Using the Edits Manager
- Defining and Maintaining Registry Edits
- System Task to Execute Edits in Patient Sets
- System Reports Related to Edits
- Color Codes Used in SEER\*DMS Editors

### Understanding Edits in SEER\*DMS

The components of SEER\*DMS edits are described below.

#### Edit ID

A unique identifier assigned to each edit. You must specify an ID for a new registry edit (1-50 alphanumeric characters may be used).

### *Guidelines for specifying an ID for a registry edit:*

- Use your registry's two-character abbreviation as a prefix for all registry-defined edits (AK, CN, CT, DT, HI, IA, LA, NM).
- If you are creating an edit designed to replace or supplement a SEER Edit, the ID should consist of your registry's prefix followed by the SEER Edit ID (e.g., DT\_IF29, IA\_IF02).

### **Group**

In SEER\*DMS, edits are organized into five groups:

1. SEER Edits validate fields submitted to SEER. These edits were developed and are maintained by the developers of the SEER\*Edits software.
2. SEER Extended Edits are defined and maintained by SEER for the SEER\*DMS registries. These edits validate fields that are supported by SEER\*DMS, but not required to be transmitted to SEER.
3. Registry Edits are defined and maintained by registry staff. These would include edits to validate registry-specific fields, edits to implement validation not covered by SEER edits, and registry-specific versions of SEER edits (the registry may decide to vary the logic of a SEER edit by creating a registry-defined edit and disabling the SEER edit).
4. SEER\*DMS edits enforce database integrity of fields common to all SEER\*DMS registries
5. SEER\*DMS Registry edits enforce database integrity of registry-specific fields. The SEER\*DMS development team creates these edits, if necessary, when registry-specific fields are added to the database.

### **Entities**

The entity defines when an edit will be executed by defining the data to which the edit will be applied. The available entities include complete database tables, for example, some edits may be run against all patient sets (entity = patients). Other entities are subsets of database structures (for example, edits may be run against "CTCs with sequence = 1" or "CTCs needing review").

Registry staff may not add or remove entities to the list. The list of available entities is defined by the SEER\*DMS development team when the system is configured for the registry.

### *Guidelines for specifying Entities for registry edits:*

- Record Edits: Select an entity related to Records ("Records" may be the only appropriate entity or your registry may have entities that define subsets of records).
- Patient Set Edits: Based on the fields used in the edit's logic, select an entity related to the database table at the lowest level of the patient set hierarchy.
  - Identify the database table used for each field used in the logic of the edit.
  - Refer to the Simplified DB Model of the Patient Set provided in the Database Documentation section of the SEER\*DMS Web Portal.
  - Determine the lowest level table used in the edit's logic.
  - Select an entity related to that table.
  - For example, an edit comparing age at diagnosis and age at birth uses the CTC and Patient tables. The CTC table is at a lower level in the patient set hierarchy than the patient table. Therefore, you should select an entity related to the CTC table.
- If your edit requires an entity that is not listed, submit a change request via the SEER\*DMS Technical Support Squish project.

## Message

The edit message is displayed in the patient set or record editor when the edit fails. The Edit ID and Message are shown when you hover over a field involved in the failed edit. The edit messages are also displayed in all edit reports. The maximum length for an edit message is 500 characters.

*Guidelines for defining Messages for registry edits:*

- The edit message should reflect the logic of the edit and should be written in language that is easily understood by data editors.
- It is recommended that you include “review required” in the message text if the edit is associated with a review flag.

## Severity

Each edit is assigned a severity level of low, moderate, high, or critical. The severity level is used to determine whether the data can be saved, whether a manual task is required, and to prioritize the edits for the user performing editing tasks.

Record edits are assigned a high severity level if the data item is required in order to screen the record for reportability or special studies. Therefore, a record edit with a high severity level will trigger a Resolve Record Errors task. If a record only has fields with low or moderate edit errors, the fields are flagged but the record moves forward in the workflow and is screened for reportability. If a record has one or more fields with high or critical edit errors, the record is sent to a manual Resolve Record Errors task for review prior to reportability screening.

Edits with a critical error level must be resolved in order to save the record or patient set. The critical error level is typically reserved for edits that check values to determine if they violate database constraints. For example, a critical edit error would occur if an “A” were entered in a field that requires a numeric value. *Note: It is technically possible to configure the edits to use the critical error level for other purposes. However, it is recommended that you only assign the critical error level to edits that check for database constraints.*

*Guidelines for setting the severity level of a registry edit:*

- You should rarely define an edit as critical. Typically, the SEER\*DMS development team will define a critical edit, if necessary, when a registry-specific data field is added to the database. Remember that your editing staff will not be able to save their changes if the changes generate a critical error.
- Record Edits:
  - Set the severity level to High if you want SEER\*DMS to create a Resolve Record Errors task when a record fails this edit.
  - Set the severity level to High if the edit is tied to the record reviewed flag for the purpose of initiating follow-back. A manual Resolve Record Errors task will provide a user with the opportunity to create the follow-back, set the reviewed flag, and save the record. The record would then move forward in the workflow.
  - If you want the edit to flag fields as having a problem but do not want a Resolve Record Errors task created, set the severity level to either low or moderate. Registry policy dictates when to use low versus moderate.
- Patient Set Edits:
  - Registry policy dictates whether to assign a severity of low, moderate, or high to a patient set edit.
  - For SEER and SEER\*DMS edits, the low severity level typically indicates that a review flag will over-ride the edit.

- The severity levels of failing edits do not affect the workflow routing of patient sets. If a patient set is saved with edit failures in a Visual Edit Patient Set or Consolidate task it will move to a Resolve Patient Set Errors task, regardless of the severity level of the failing edits.

## Dependencies

SEER\*DMS uses “dependencies” to conditionally execute an edit based on the success of other edits. Typically, a dependency is added to an edit to ensure that a valid value is provided for a required field. For example, several SEER edits require age at diagnosis. These edits are not executed if the “*age\_at\_diagnosis*” edit fails.

There are two sets of dependencies for each edit. An edit is not executed if any of the edits listed in its “Depends on” section fail. Similarly, if the edit fails then none of the edits in the “Depended on by” section will be applied.

To view examples, review the dependencies for the following SEER edits: Age\_at\_diagnosis (Age at diagnosis is not valid), IF118 (Age, Primary Site, Histology conflict), IF13 (Age, Birth Date and Date of Diagnosis conflict).

Dependencies can only be defined between edits within the same group. For example, a registry edit can only depend on registry edits and cannot depend on SEER edits.

## Groovy Code

SEER\*DMS edits are implemented in Groovy, a scripting language for the Java platform. The Internet has several Groovy references including the Groovy home page at <http://groovy.codehaus.org>. You do not need to be fluent in Groovy to add or modify edits. SEER\*DMS edits use a small subset of the Groovy syntax. A working knowledge of regular expressions and Groovy logic statements are all that is needed to add and modify edits in SEER\*DMS. To define a new edit, it is recommended that you copy-and-paste the definition from an existing edit and use that code as a template.

*Guidelines for writing the Groovy code for a registry edit:*

- An edit error is triggered if the code returns FALSE for the record or patient set. The edit passes if the code returns TRUE.
- Use the Groovy code of a similar edit as a template.
- If you copy-and-paste the Groovy code of a SEER, SEER\*DMS, or SEER Extended edits, verify that an appropriate entity is available. If your edit requires an entity that is not listed, submit a change request via the SEER\*DMS Technical Support Squish project.
- The Available Contexts are displayed at the bottom of the page. Your Groovy code may include references to any of the Available Contexts.

## Contexts

A context is a Java naming system. In SEER\*DMS, contexts are used to define arrays, hash tables, and functions used by the edits. For example, there are a large number of contexts defined for the SEER\*Edits. Primarily, these represent data tables required by the SEER Edits logic. Registry staff may not add or modify contexts in the current version of SEER\*DMS.

## Lookup Queries

The edits use queries to verify values against tables in the database. These are primarily used to query lookup tables in the database (lkup prefix). Typically, the edit lookups simply check to see if a code exists in the database table, but other logic may be incorporated. The full query for each lookup can be seen in the list of Lookups shown on the Definition tab of the Edit Editor. Registry staff may not add or modify lookups in the current version of SEER\*DMS.

## Documentation

The documentation section contains descriptive text that is more comprehensive than the edit's message. If available, this documentation is displayed in the edit popup shown in the patient set and record editor. Documentation provided by the SEER program is included for the SEER Edits. Registry staff are responsible for writing and maintaining documentation for registry edits.

## Viewing Edit Documentation

SEER\*DMS provides four ways to view documentation related to edits:

- The SEER\*DMS editor allows you to view documentation for edits that are failing in the record or patient set. If you are using a wide monitor resolution, you may view the Edits in the panel displayed on the right side of the screen. If your resolution does not support the right-side panel, you may view them on the Edits page within the editor. The full documentation and Groovy code can be viewed by clicking the information icon (you may toggle the view between definition and source code).
- All system users have access to the Help menu which includes an Edits section. You may use the Edits Help page to search for an edit by edit ID, group, severity level, and/or search text (see the *Using the Edits Help Page* section of this chapter for instructions).
- If you have the *system\_administration* permission you may use the Edits Manager to view documentation and source code for all edits. You may also use the Edits Manager to modify registry-defined edits. (See the *Using the Edits Manager* and *Defining and Maintaining Registry Edits* sections of this chapter for instructions.)
- If you have the *system\_administration* permission, you can search the XML configuration files in which the SEER, SEER Extended, and system edits are defined. Refer to the *System Administration Page* section of Chapter 27 for instructions to access the system files. The edit system files are named according to the edit group. Once you select the XML file, use the Firefox search tools to find a particular edit.

## Using the Edits Help Page

Requires system permission: *none* (all users have access to the help pages)

To search the Edits Help page for documentation related to an edit:

1. Select **Help > Edits**. By default, all edits will be listed.
2. Use the drop-down filters to search by edit group, data type (entities), and severity level.
3. Enter text in the *keywords* box to search for text in the edit message, ID, or Groovy source code.
4. Click **Filter** to apply the search criteria; or **Clear** to reset the filter to the default values.

The following example shows the result of using the filter to find Edit Group = SEER, Data Type = All, Severity Level = All, and search text = *age\_at*.

<All Groups> <All Data Types> <All Severities> age\_at Filter Clear

**Age\_at\_Diagnosis** Moderate SEER

Age at diagnosis not valid.


Click the edit's ID to expand its help section, as shown below. The edit's Message is listed first, followed by Entities, Dependencies, Documentation, Groovy Code, and History (if available). Each component is described in the *Understanding Edits in SEER\*DMS* section of this chapter.

## Using the Edits Manager

Requires system permission: *system\_administration*

Use the Edits Manager to search for an edit or to review a set of edits. If you are modifying or deleting an edit, refer to the *Defining and Maintaining Registry Edits* for additional instructions.

To filter the list of edits in the Edits Manager:

1. Select **System > Edits**. By default, all edits will be listed.
2. The **Edit ID** is the edit's name or label (examples: IF104, SEER\_IR09, Age\_at\_Diagnosis). Use the **Edit ID** filter to search for an edit based on a complete or partial Edit ID.
3. Use the **Group** filter to limit the list to the edits in a particular group. The edit groups are described in the Understanding Edits in SEER\*DMS section of this chapter.
4. Use the **Severity** filter to search for edits based on severity level. Each edit is assigned a severity level of low, moderate, high, or critical. Please refer to the *Understanding Edits in SEER\*DMS* section of this chapter for a detailed explanation of edit severity levels.
5. Use the **Properties** filter to search for edits which reference a specific database field in the Groovy source code. You may click the Lookup icon  to search for a field; or enter text directly into the filter (SEER\*DMS will display a list of properties that match your text as you type). The text used in the Properties lookup is the Description shown on the Field Mappings help page.

6. Use the **Data Type** filter to search for edits based on the tables used in the edits' Entities.
7. Use the **Info** filter to search by text included in the message, ID, or source code.
8. If the filters are set appropriately, click **Apply**.
9. Click the ID of an edit to view or modify the edit's logic, message, or documentation.
10. If you would like to restore the default filter settings, click **Reset**.

# Defining and Maintaining Registry Edits

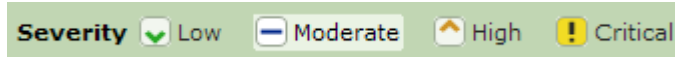
Requires system permission: *system\_administration*

The instructions below provide an overview of the steps required to add, delete, or modify an edit. Refer to the *Understanding Edits in SEER\*DMS* section of this chapter for specific guidelines when selecting values for each component of the edit.

To create or modify a registry edit:

1. Select **System > Edits**.
2. If you are creating a new edit, click **Add Edit**. If you are modifying an existing edit, use the filters to find the edit and then click the edit's **ID**.
3. Set all components shown on the **General** tab (specific guidelines are provided in *Understanding Edits in SEER\*DMS*).

- a. Enter or modify the **Edit ID**.
- b. Select one item listed in the **Entities** drop-down list.
- c. Click the **Severity** level that you wish to assign to this edit. Highlighting is used to identify the current selection. In the example below, the Severity level is Moderate.



- d. Enter or modify the **Message** text.
  - e. An edit is not executed if any edit listed in the **Depends on** section fails. To add an edit to the Depends On section:
    - i. Click **add dependency**
    - ii. Use the filter to search for the edit. Dependencies can only be defined between edits within the same group. Therefore, only registry edits will be listed.
    - iii. Click the **Edit ID**.
  - f. If you wish to remove an edit from the Depends On section, click the **remove** link associated with that edit.
4. On the **Definition** tab, enter or modify the **Groovy** source code that defines the logic of the edit. Your Groovy code may include references to Contexts and Lookups. You may view the available Contexts and Lookups using the links at the bottom of the page. Contexts are organized by the edit groups. A Context is directly available to edits within the Context's group. However, edits in other groups may reference Contexts in other groups. You must specify the group in the getContext call. Example of using a context (*repYearByRegistryId*) from another group (SEER):

```
def yearByRegistry =  
    Functions.asInt(Functions.getContext("seer", "repYearByRegistryId")[registryId]);
```
  5. On the **Documentation** tab, enter detailed documentation related to the new edit or modify the documentation to reflect the changes that you made to the logic.
  6. Click **Save** to exit and save your changes.

To delete a registry edit:

1. Select **System > Edits**.
2. Use the filters to find the edit as described in the *Using the Edit Manager* section of this chapter. To search for registry-defined edits, set the Group filter to your registry's name. (SEER, SEER Extended, SEER\*DMS, and SEER\*DMS Registry edits cannot be deleted.)
3. Click the edit's **ID**.
4. Review the list of edits that depend on this edit (listed under the Depended on By heading). Determine whether this edit must be retained in order for those edits to function properly; or if the listed edits require modifications due to this edit being deleted.
5. If you determined that the edit can be deleted, **delete edit**. (This link is shown in the upper right corner of the screen for Registry Edits. It is not displayed for SEER, SEER Extended, SEER\*DMS, or SEER\*DMS Registry edits.)

The screenshot shows the SEER\*DMS Edit Editor interface. At the top, the user is identified as 'coyle'. The main content area is titled 'Edit Editor' and has tabs for 'General', 'Definition', 'Documentation', and 'History'. The 'General' tab is active. The 'Edit ID' is 'IA001' with a 'change id' link. The 'Group' is 'Iowa'. The 'Entities' dropdown is set to 'Patients'. The 'Severity' is set to 'Moderate'. A message box contains the text: 'Please review all patient demographic information. Demographic Review Flag must be set to 1 to confirm completion.' Below the message, there are 'Depends on' and 'Depended on by' sections. At the bottom, there are 'Save' and 'Cancel' buttons. A 'delete edit' link is located in the top right corner. Two callout boxes provide additional information: one points to the 'delete edit' link, stating 'Click this link to delete a Registry-defined edit. This is not available for SEER, SEER Extended, SEER\*DMS, or SEER\*DMS Registry edits.' The other points to the 'Moderate' severity level, stating 'Moderate is the Severity level for this edit. An edit's severity level is shown in a highlighted box.'

## System Task to Execute Edits in Patient Sets

All edits are executed each time a patient set is opened, validated, or saved in the SEER\*DMS editor. The Patient Set Edits system task enables you to re-execute the edits on patient sets in the database. You may run the edits on all patient sets or on a cohort defined by year of diagnosis. Use the Patient Set Edits task to ensure that new or modified edits are evaluated.

A polisher is a system utility that derives, calculates, or assigns data field values. For example, polishers are used to derive collaborative stage variables; assign census tract based on address; and calculate the age at diagnosis based on date of birth and date of diagnosis. When a patient set is opened, saved, or validated, a polisher will be executed if the value of a related data item changes. Occasionally, you may need to execute a polisher on a large number of patient sets. The Patient Set Edits task enables you to run selected polishers as well as the edits.

Polishers are assigned to "classes". Classes are simply categories used in SEER\*DMS to define when the polishers are executed during various system processes. A polisher may belong to multiple classes. The Polishers help page lists the polishers and the classes to which they belong.

The Patient Set Edits task has an option to execute the Edit Polishers. Edit polishers are polishers in the "Pre-Edits" and "Post-Edits" classes (the polishers in these classes are listed on the screen when you open the Patient Set Edits system task). These are polishers that need to be run with the Patient Set Edits task during the initial migration of data to SEER\*DMS. The edit polishers rarely need to be run as a group after deployment.

A polisher may need to be executed with the Patient Set Edits task when a new version of SEER\*DMS is released that includes a change to that polisher. The Patient Set Edits task allows you to select one to three “extra” polishers. All of the standard polishers (polishers in the “Validation” class) can be executed with the Patient Set Edits task.

*To re-execute the edits for some or all patient sets in the database:*

1. Click **System > Tasks**.
2. Click the **Patient Set Edits** link.
3. To limit the edits to data by year of diagnosis, enter the **Start Year**. Patient sets with a diagnosis date during or after this year will be considered.
4. If you would like to define an end range, enter an **End Year**. If the end date is not specified, today’s date will be used by default.
5. To include data with unknown year of diagnosis, set **Include Unknown Year** to *Yes*.
6. If you wish to execute all polishers in the Pre-Edit and Post-Edit classes, set **Run Edit Polishers** to *Yes*. It is recommended that this option be set to *No* unless there is a specific need related to the transitioning of data into SEER\*DMS.
7. You may execute one to three individual polishers with the Patient Set Edits using the **Extra Polisher** drop-down menus. These are polishers in the Validation class (as listed on the Polishers help page).
8. You may enter text related to this task in the **Comment** field. The comment for the last execution of the task is stored in the database (utility\_history table).
9. Click **Start**.

The edits will be re-evaluated for each patient set in the cohort. In order to avoid creating an inordinate number of worklist tasks, a Resolve Patient Set Errors task will *not* be created for each patient set with an edit error. If the logic of a new or modified edit is implemented incorrectly, it could erroneously create an edit error for a large number of patient sets. Therefore, you must use reports rather than tasks to identify the patient sets with errors and to evaluate the error levels in the patient set data. Two system reports, RPT-064A and RPT-064B, are available for identifying the edit errors that were triggered and the patient sets that are involved:










## System Reports Related to Edits

Report ID	Title	Description
RPT-064A	Frequency of Edit Errors in Patient Set Data	Use this report to evaluate the error levels in the Patient Set data
RPT-064B	Patient Sets with Edit Errors	A listing of Patient Sets with a failure related to a particular edit or a sample for all failing edits.
RPT-064C	SEER*DMS Edits	ID, Message, Severity, and Group of all edits available in SEER*DMS.
RPT-064D	Detailed Listing of SEER*DMS Edits	Lists all components of edits, including the Groovy code, entities, history, and documentation.
RPT-064E	Number of Edit Failures per Patient Set	Lists the Patient Set ID and the number of failed edits for each Patient Set with edit errors.

## Color Codes Used in SEER\*DMS Editors

The SEER\*DMS record and patient set editors display data fields in different colors to highlight fields that contain errors, and fields that have been modified but not validated or saved. Although you can readily see the fields that require attention on data pages, it is recommended that you review the list of errors and related data fields on the Edit Errors page prior to making changes to a record or patient set. Refer to the chapter related to your specific task for further instructions.

If two or more edits are related to a field, the edit with the highest severity level will determine the color.

	<b>White</b> indicates that the field does not have an error and has not been changed since the last save.
	<b>Dark red with white text</b> indicates that the field triggered a <b>Critical Error</b> . The value can not be stored in the system due to database constraints. Critical Errors will only be seen in patient sets, since database constraints are not applicable to record data.
	<b>Medium red with black text</b> indicates that the field triggered an error with a <b>High</b> severity level. In patient set data, the value may not be valid for this field or an inter-field edit may have detected a conflict with other fields. In record data fields, this indicates that an error was detected that must be resolved prior to screening. The record will not move past the Resolve Record Errors task until all errors with a severity level of high are resolved.
	<b>Light red</b> indicates that the field has an error with a <b>Low</b> or <b>Moderate</b> severity level. The value may not be valid for this field or an inter-field edit may have detected a conflict with other fields. Low and moderate errors should not be resolved when editing records, these editing tasks should be performed when consolidating the patient data.
	<b>Orange</b> indicates that a field with an error was modified, but the change has not been validated against the SEER and local edits. Fields are validated when the record is saved or the Validate button is clicked. Once validated, this field will turn to yellow if the error was corrected or to a shade of red if the field still contains an error.
	<b>Yellow</b> indicates that the field that did not have an error and was changed. The patient set or record has not been saved since the change was made. Once validated, this field will turn to a shade of red if there is an error.
	<b>Blue with a black border</b> indicates that the field was changed by a Polisher.
	<b>Blue with no border</b> indicates that the field is read-only and was changed by a Polisher. <i>Note: Some fields are modified when another field is changed.</i>
	Read-only fields that have not been changed are shown in a lighter shade (the color is determined by the color scheme that you are using).