Managing Data

Managing the ServiceNow Database

Introduction to Managing Data

Note: *This article applies to Fuji and earlier releases. For more current information, see Data Management* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

Overview

Data in ServiceNow is stored and managed according to a principled structure that administrators can view and configure.

- The data is stored in database tables and records.
- Various tools exist to manage this data.
  - The schema map displays CI relations graphically.
  - Data dictionary tables store structure and relationship definitions.
- Various plugins provide additional functionality.
- Data import and export simplifies managing large sets of data through import maps and a number of exportable formats.
- Field normalization makes records more readable and prevents duplication of data.
- Scripting support allows for advanced customizations and automation of data management.

Database Structure

All of the information in the instances is stored in *tables*, which consist of a series of *records*. The record in turn holds a series of *fields* that hold the individual bits of data and can be viewed either as a list or a form.

Tables can be related to each other in the following ways:

- **Extensions:** A table can extend another table. The table doing the extending (child class) includes all of the fields of the other table (parent class) and adds its own fields. For instance, the Incident [incident] table has all of the Task [task] table fields (because an incident is a special form of task) and has its own incident-specific tasks. For more information, see Tables and Classes.
- **One-to-Many:** Within a table, a field can hold a reference to a record on another table. There are three types of one-to-many relationship fields:
  - *Reference Field:* allow a user to select a record on a table defined by the reference field. For instance, the **Caller** field on the Incident table allows the user to select any record on the User table.
  - *Glide List:* allows a user to select multiple records on a table defined by the glide list. For instance, the **Watch list** field on the Incident table allows the user to select records on the User table.
  - *Document ID Field:* allows a user to select a record on any table in the instance. These fields are much less common, but one example is the **Document** field on the Translated Text [sys_translated_text] table.
- **Many-to-Many:** Two tables can have a bi-directional relationship, so that the related records are visible from both tables in a related list. For more information, see Creating a Many-to-Many Relationship.
- **Database Views:** Two tables can be joined virtually with Database Views to enable reporting on data that might be stored over more than one table.

Data Management Tools

There are a number of tools that can help manage data within the instance:

- **Schema Map:** displays the relationships between tables visually, helping to navigate through the database structure.
- **Data Dictionary Tables:** holds information that defines the database and can be accessed for information on the database schema.
- **Table Cleaner:** automatically deletes records on certain tables to prevent data growing exponentially.

Schema Map

The Schema Map provides an interface for viewing the relationships between tables. The inter-table relationships it captures include many-to-many relationships, tables that extend other tables, and tables that reference other tables through reference fields.

Data Dictionary Tables

These tables hold important information on the database and its structure:

- **Tables [sys_db_object]:** contains a record for each table. (Prior to the Calgary release, contains information on how tables extend one another and is called the Table classes table.)
- **Dictionary Entries [sys_dictionary]:** contains additional details for each table and the definition for every column on each table. Each row represents either a column on a table or a table.
- **Field Labels [sys_documentation]:** contains the human-readable labels and language information.

For more information, see Data Dictionary Tables.

Table Cleaner

The system automatically deletes records from specific tables after a specific time to deletion. Deleting these records automatically prevents the tables from growing to an unmanageable size. The time before a record is deleted begins on the date and time value in the tracked field.

The Table Cleaner scheduled job runs the table cleaner every hour. To view the list of tables that are auto-cleaned, go to

```
https://<yourInstanceName>.service-now.com//sys_auto_flush_list.do
```

All records with [**MatchField** < (**current_time** - **Age in seconds**)] are deleted.

The **MatchField** field represents a Date/Time column in the table that you are trying to clean up.

The **Age in seconds** field represents a value in seconds.

You can set up multiple table cleaner entries for a particular table. Performance depends on the size of the table and the conditions used. For example, if you use a custom column in a very large table that has no index on, performance is severely degraded. Performance also depends on the number of rows to be deleted.

Several dictionary attributes available with the Calgary release control the behavior of the table cleaner.

Data Management Plugins

Furthermore, there are a few plugins that extend the data management functionality:

- **Data Archiving plugin:** provides the ability to archive records to minimize performance issues.
- **Database Rotations plugin:** provides tools for managing large tables to minimize performance issues.
- **Discovery plugin:** provides the ability to auto-populate the CMDB with data about hardware, software, and networks.
- **Domain Support plugin:** provides the ability to separate an instance into different distinct domains.
- **Many to Many Task Relations plugin:** provides the ability to define many-to-many relationships between task tables.

Database Rotations Plugin

The Database Rotations plugin mitigates performance issues caused by large tables by breaking up overly-large tables into a series of smaller tables, through two methods:

- **Table Rotation:** sets up multiple tables for the same purpose that the platform uses on a rotating basis, deleting old information as it goes.
- **Table Extension:** creates new tables for the same purpose as the platform fills old tables.

Many to Many Task Relations Plugin

The Many to Many Task Relations plugin allows administrators to manage the many-to-many relationships between task tables and record the relationship in more detail beyond Parent > Child.

Domain Support Plugin

The Domain Support plugin provides a method for separating an instance into different domains, including partitioning data and securing the domains so that only members of the domain can see the data from their domain. The Data Separation part of the plugin can be implemented separately, if desired.

Importing and Exporting Data

The following tools are available to export data from an instance:

- Manually exporting from a record list.
- Exporting data on a schedule with scheduled reports.
- Exporting data with a URL parameter.
- Exporting data with a .NET application.

The following tools are available to import data from an instance:

- Manually importing from a record list.
- Importing data with import sets.
- Accepting external data onto the ECC queue.

In addition, data can be imported or exported with web services. LDAP data can be accessed with the LDAP integration in the base system. For more information on integrating with other platforms, see Integration Overview.

Configuring Import and Export Properties

You can use the Import Export Properties page to configure import and export properties, starting with the Fuji release. To access the page, navigate to **System Properties > Import Export**. The page lists properties categorized by either import or export, and by data format such as CSV or XML. The **Current Value** appears in black if a custom value is set, or in grey if no custom value is set.

Field Normalization

Field Normalization includes two features: *normalization* and *transformation*.

- **Normalization** forces the ServiceNow platform to convert different forms of the same field value to a single, accepted value automatically. By forcing a field to use a simple, recognizable description for multiple variations of the same thing, normalization can eliminate duplicate records and make searches easier. In addition to reconciling different forms of the same value in fields, normalization can be configured to adjust queries automatically to return normalized results.
- **Transformation** enables an administrator to transform raw field input into standardized values that are more meaningful to an organization. An example of a standardized value might be to round RAM size in configuration items to a whole number, such as 4000 MB instead of 4112 MB. Transformations are controlled by parameters and conditions and can be configured to return transformed values in queries.

Managing Data through Scripts

ServiceNow provides a robust platform for managing and manipulating data through scripts. The primary API for querying and manipulating data in tables within an instance is GlideRecord. For more information on scripting in the platform, see Glide Stack.

Enhancements

Fuji

- You can configure import and export properties using a graphical interface.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/managing-data/concept/c_DataManagement.html

Data Dictionary Tables

Data Dictionary Tables

Note: *This article applies to Fuji. For more current information, see Data Dictionary Tables* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

These tables provide data dictionary, data modeling, and entity relationship information:

- **Tables [sys_db_object]:** contains a record for each table.
- **Dictionary Entries [sys_dictionary]:** contains additional details for each table and the definition for every column on each table. Each row represents either a column on a table or a table.
- **Field Labels [sys_documentation]:** contains the human-readable labels and language information.

Dictionary Entries

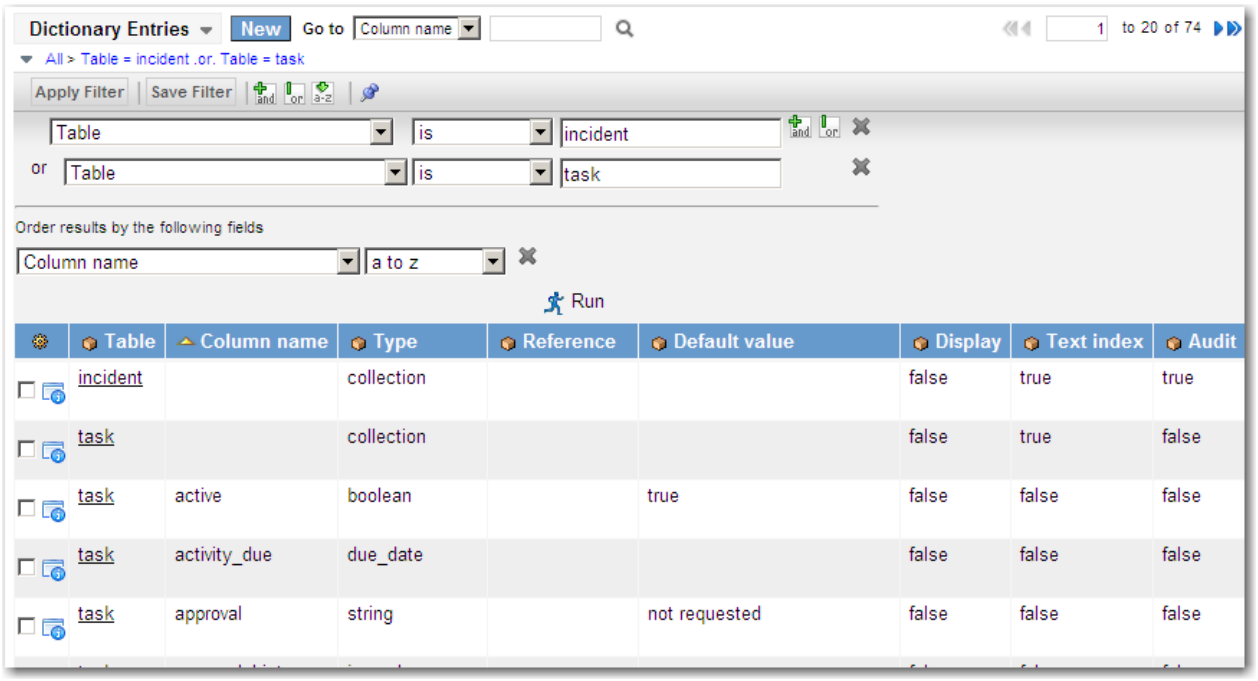
The Dictionary Entries [sys_dictionary] table, also called the **System Dictionary**, defines every table and field in the system. It contains information about a field's data type, character limit, default value, dependency, and other attributes.

Access the system dictionary in one of these ways:

- To see the system dictionary list view, navigate to **System Definition > Dictionary**.
- To view particular dictionary definition, right-click the list header, form header, or field label, and select **Configure Dictionary** (**Personalize Dictionary** in versions prior to Fuji).

To learn more, see System Dictionary.

The following image shows a filtered list of dictionary entries for the Incident table and the Task table, which it extends.



Field Labels

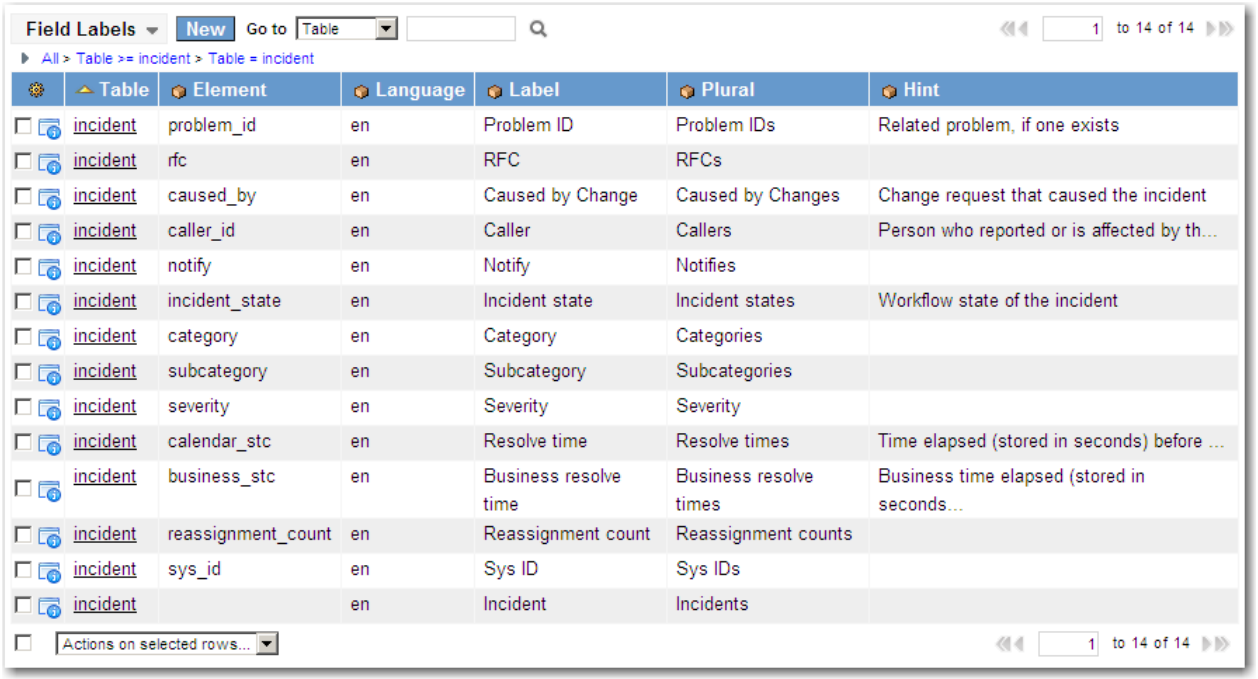
The Field Labels [sys_documentation] table, also called the **Language File**, contains information about the labels and hints for each table and column in the system.

Access the language file in one of these ways:

- To see the list view, navigate to **System Definition > Language File**.
- To see the field label for a particular field, right-click the field label on the form.

For more information on the language file, see Relabeling a Table or Field.

The following image shows the language file filtered to display only labels on the Incident table.

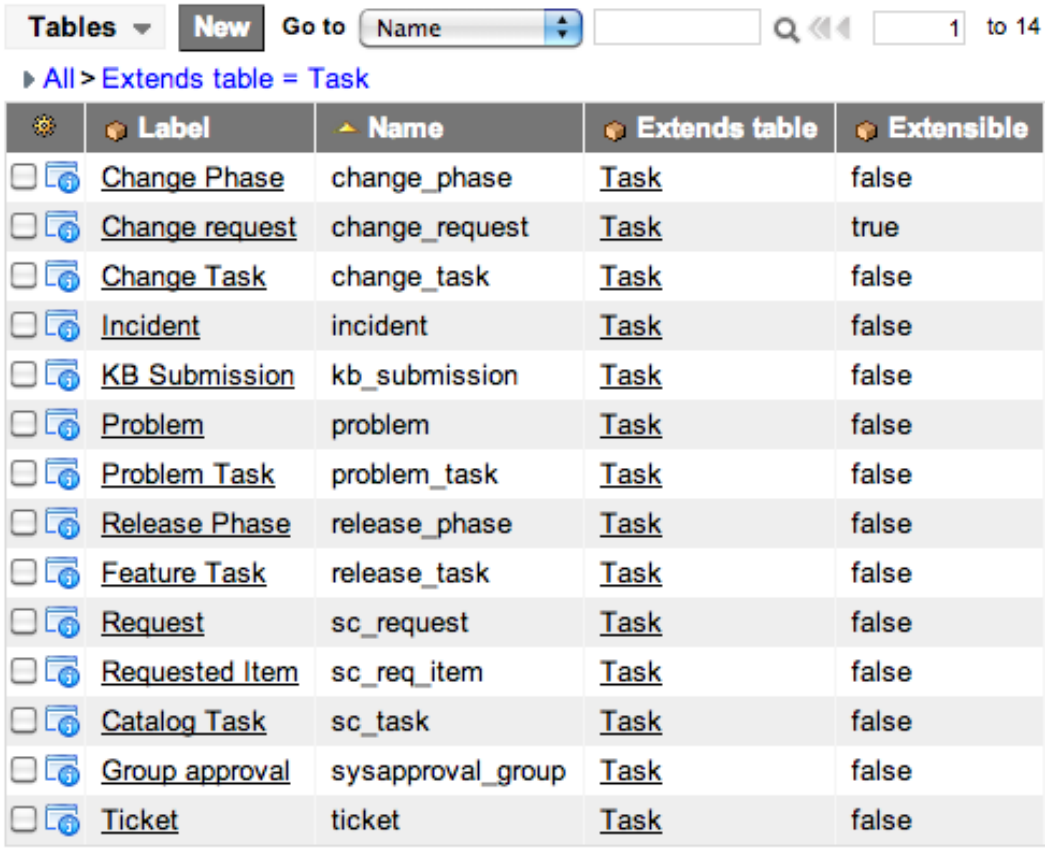


Tables

The Tables [sys_db_object] table contains a record for each table in the database (Calgary release).

Access the Tables list by navigating to **System Definition > Tables**. Administrators can create a custom table, add or modify columns in a searchable and sortable embedded list, and define the auto-number format. To learn more, see Creating a Custom Table.

The following image shows a list of the tables that extend the Task table.



Label	Name	Extends table	Extensible
Change Phase	change_phase	Task	false
Change request	change_request	Task	true
Change Task	change_task	Task	false
Incident	incident	Task	false
KB Submission	kb_submission	Task	false
Problem	problem	Task	false
Problem Task	problem_task	Task	false
Release Phase	release_phase	Task	false
Feature Task	release_task	Task	false
Request	sc_request	Task	false
Requested Item	sc_req_item	Task	false
Catalog Task	sc_task	Task	false
Group approval	sysapproval_group	Task	false
Ticket	ticket	Task	false

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/managing-data/concept/c_DataDictionaryTables.html

System Dictionary

Note: *This article applies to Fuji. For more current information, see System Dictionary* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

The system dictionary is a table, called Dictionary Entry [sys_dictionary], that contains details for each table and the definition for every column on each table in an instance. Each row in the system dictionary represents either a table or a column in one of the tables. The system dictionary provides options for administrators to modify tables and fields, which in turn define lists and forms.

Altering System Dictionary Records

Use caution when changing system dictionary records because changes can have a high impact on functionality. In particular, changes to dictionary entries for system tables, which are tables that begin with **sys_**, can create system-wide issues such as the inability to use update sets.

Dictionary changes are difficult to reverse. Also, dictionary changes automatically apply to all extended tables unless a dictionary override is defined. Be sure that changes are well-tested before applying them to a production instance.

Creating Dictionary Entries

In most cases, use the following interfaces rather than creating entries directly on the system dictionary:

- To create new tables and fields, use the Tables module.
- To create new fields, configure the table form.

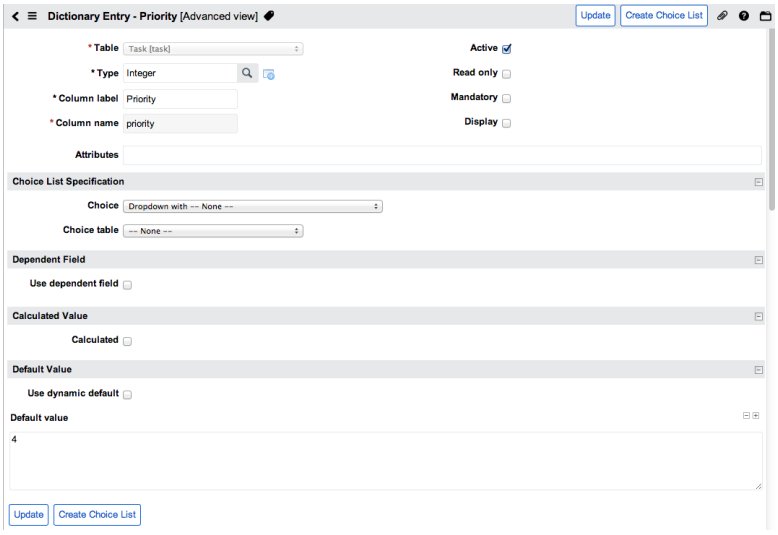
When you create a field from the system dictionary, it is automatically added at the end of the first section of the default form view.

Modifying Dictionary Entries

To modify a dictionary entry, open the record and fill in the fields on the form, as appropriate (see table). You can open a dictionary entry record in any of these ways:

- Through a form:
  1. Navigate to a field in a form or list.
  2. Right click a field and select **Configure Dictionary** (**Personalize Dictionary** in versions prior to the Fuji release) or **Show [fieldname]**. The system dictionary entry for the field opens.
- Through the Dictionary module:
  1. Navigate to **System Definition > Dictionary**.
  2. Click an entry for a field or table. Entries for tables have **Type** set to **Collection**.

The Dictionary Entry form provides an Advanced view and additional fields starting with the Eureka release.



Advanced view of the dictionary entry for the **Priority** field on the Task table

Field	Description
Table	Defines the table in which the element is created. **Note:** This list shows only the tables that meet the scope protections for adding fields (starting with the Fuji release).
Type	Defines the field type of the column or, if the **Type** is **Collection**, indicates that the dictionary entry represents the table. For more information about field types, see Introduction to Fields. You cannot change a field's type without data loss. For custom fields, the system prompts you to confirm you accept the data loss before proceeding with the change. For a base system field, the system prevents you from making the type change.
Active	Enables or disables the field. When this check box is cleared, the field is not used by the system.
Read only	Determines whether users can change the field value. When this check box is selected, users cannot change the value. The data for the field is calculated and displayed by the system. **Note:** You can override this option for extended tables.
Audit	Enables or disables auditing for a table. For more information, see Turning on Auditing (History) for a Table. **Note:** This option only applies to tables. To exclude fields from auditing, see Excluding an Individual Field from Auditing.
Text index	Determines whether searches index the text in a table. For more information, see Enabling Text Search in Record Lists. **Note:** This option only applies to tables. To exclude fields from indexing, see Removing an Index for a Specific Field.
Column label	Defines a unique label for the column. The label appears on list headers and form fields for the column. • Updating the **Column label** field also updates the label in the language file (for the current language). For more information, see Field Labels. • When you create a new column, the column name is populated automatically based on the label, which is prefixed with **u_** to indicate that it is custom. For example, if you enter **Activity Description** as the column label, the column name defaults to **u_activity_description**.
Column name	Defines the field name of the column. When you create a new field, this name is populated automatically based on the label and a prefix as follows: • For a field on a table in a different scope, the name is prefixed with the scope to indicate that it is custom and not part of that application (starting with the Fuji release). • For a field on a table in the same scoped application, the name does not have a prefix, which indicates that it is part of the application. • For a field in a global application and for version prior to Fuji, the name is prefixed with **u_** to indicate that it is custom. You cannot modify the prefix; however, you can modify the rest of the name. The name can contain only lowercase, alphanumeric ASCII characters and underscores (_). You cannot change the name of an existing dictionary record.

Max length	Provides a logical limit for the size of string fields to determine how the system displays them in the user interface and how to map them to physical database data types. String fields with a length under 255 characters appear as a single-line text fields. String fields with a length over 254 characters appear as a multi-line text box. The system maps the field length to the closest physical data type available on the database. In some cases this results in significantly more available length than originally specified. For example, entering a length of 50 maps to the closest physical data type of VARCHAR(100), which provides up to a 100 character limit or double the requested field length. Likewise, entering a length of 1000 maps to the closest physical data type of MEDIUMTEXT, which provides up to a 4000 character limit or four times the requested field length. See Database Field Types for MySQL physical data types. **Notes:** • You can only change this value for a **String** field. Changes for any other type of field are ignored. • Users on an Oracle instance cannot increase the maximum length of a string field to anything greater than 4000 through the application UI as this requires the CLOB datatype in Oracle. To increase beyond this size, log an incident with Technical Support to request the change. • To prevent data from being lost, only decrease the length of a string field when you are developing a new application and not when a field contains data. A warning appears if a change to a custom field will result in data loss. For a base system field, you cannot make a change that will result in data loss.
Mandatory	Determines whether this field must contain a value to save a record. For more information, see Creating Mandatory Fields. **Note:** You can override this option for extended tables.
Display	Indicates that this field is the display value for reference fields. Set this to true for the one field whose value you want to use as the text displayed in links to this table on lists and forms. By default, the **Number** field is the display value for all task tables. **Note:** • This option does *not* control whether a list or form displays this field as part of the layout. Instead, see Configuring Lists and Configuring Forms. • The display value becomes part of the form title when viewing an individual record from a table. • You can set a different display value on an extended table than the display value on a parent table by using a dictionary override.
Attributes [Advanced view]	Alters the behavior of a field or functionality that depends on the field. For more information, see Dictionary Attributes. You can also configure attributes for this dictionary entry through the **Attributes** related list. **Note:** You can override attributes for extended tables.
Default Value	
Use dynamic default [Advanced view]	Allows you to specify a default value that is generated dynamically based on a dynamic filter.
Dynamic filter value [Advanced view]	Specifies the dynamic filter that determines the default value if the **Use dynamic default** option is selected.
Default value	Specifies the default value of the field for any new record. Ensure that this value uses the correct field type. For example, an integer field uses a default value of **2** but cannot use a default value of **two**. **Note:** You can override this value for extended tables.
Reference Specification	
Reference	Makes the field into a reference field. You can reference tables that are in the same scope or that allow read access from applications in other scopes (starting with the Fuji release). If you enter a name that does not match an existing table, a new table is created when you save your changes to the dictionary record. If the current table has a module in the application navigator, a module for the new table is automatically created in the same application menu.

Use reference qualifier [Advanced view]	Specifies the type of qualifier to use (starting with the Eureka release): • **Simple:** A set of choice lists where you can specify a reference qualifier condition. • **Dynamic:** A dynamic filter that you can use to build the qualifier. • **Advanced:** A static encoded query string or JavaScript code that you can use to build the qualifier.
Reference qual condition	Specifies a condition when the reference qualifier runs if the **Simple** qualifier type is selected (starting with the Eureka release).
Dynamic ref qual [Advanced view]	Specifies the dynamic filter that determines the reference qualifier when the **Dynamic** qualifier type is selected (starting with the Eureka release).
Reference qual [Advanced view]	Filters the records available for a reference field if the **Advanced** qualifier type is selected. For more information, see Reference Qualifiers. **Note:** You can override the reference qualifier for extended tables.
Reference key [Advanced view]	Identifies a field other than sys_id to use as the unique identifier for reference fields. For more information, see Defining the Reference Key.
Reference cascade rule [Advanced view]	Defines what happens to a record if the record it references is deleted. Select one of the following: • **Clear:** clears the references (default). • **Delete:** deletes all referencing records. • **Restrict:** prevents record deletion if there is a referencing record. • **None:** does not change referencing records. For more information, see Cascade Delete Rules.
Reference floats [Advanced view]	Enables the **Edit** button on related lists for one-to-many relationships. For more information, see Configuring Edit.
Dynamic creation [Advanced view]	For reference fields, determines whether entering a value that does not match an existing record creates a new record on the referenced table. If selected, use the **Dynamic creation script** field to define how to create the new record. For more information, see Enabling Dynamic Creation.
Dynamic creation script [Advanced view]	When the **Dynamic creation** field is selected, allows you to enter a script for creating a record on the referenced table.
Dependent Field	
Dependent on field [Advanced view]	Specifies a field on which the current field depends. For more information, see Configuring Dependent Fields. **Note:** You can override this value for extended tables.
Choice List Specification	
Choice	Allows users to see a list of suggested values in one of the following ways: • **Drop-down menu without -- None --** • **Drop-down menu with -- None --** • **Suggestion** If a choice is used, either define a choice list or use the fields **Choice table** and **Choice field** to copy choices from another field elsewhere in the dictionary.
Choice table [Advanced view]	Populates the field choices with the same values as another choice field. If the **Choice** field is set to anything besides **None**, select a table to draw choice values from. The field **Choice field** must also be populated. For example, if **Choice table** is set to the Incident [incident] table, this field has the same choice list as one of the choice fields on Incident. **Choice field** (see below) determines which field.

Choice field [Advanced view]	Populates the field choices with the same values as another choice field. If the **Choice** field is set to anything besides **None**, select a field from the table you selected for **Choice table**. For example, if the **Choice table** field is set to the Incident [incident] table, and **Choice Field** is set to **Priority**, this field has the same choices as the **Priority** field on Incident, even if those choices change. **Note:** This field must be a choice field.
Calculated Value	
Calculated [Advanced view]	Determines whether the value of the field is calculated from other values. If selected, use the **Calculation** field to define how the calculation is performed. When sorting or grouping by a calculated field, the sort order is based on the field value from the last time the field was updated, not the last time the field was displayed. **Note:** In relation to business rules, calculated fields are populated first before any business rule, even a *before* business rule, is run. Calculated fields are then populated again if necessary after any *before* business rules run.
Calculation [Advanced view]	When the **Calculated** field is selected, allows you to enter a script for calculating the value of the field. You can use the current object in this script. Just as with access control rules, the script can: • Evaluate to *true* or *false*. • Return an **answer** variable set to **true** or **false**. • Set a field value directly, such as: `current.display_name="name"`. **Note:** You can override the calculation for extended tables.
Additional fields	
Class	Identifies the table that the current table extends. Tables that do not extend other tables specify their own name in this field. For more information, see Tables and Classes.
Defaultsort	Obsolete. See default sort order.
Spell check	Enables or disables spell check on the field. For more information, see Adding Spell Check to a Field.
Unique	Requires the field value to be unique. **Warning:** Making a field unique when the corresponding table already has different values for that field causes data loss. Before you enforce uniqueness on a field, verify that no records in the table for the field have values, or that they all have the same value.
UI action	
Default view or Advanced view	Changes the form view to the default or advanced view. The fields change based on the view. If you are using the default view, you must write a script to accomplish the same tasks that advanced view fields provided.
Related Lists	
Access Controls	Provides access to the access controls that permit or limit access to the data in the table (starting with the Eureka release).
Choices	Provides access to the options in the choice list field you are editing.
Dictionary overrides	Provides access to the dictionary overrides for this field.
Attributes	Provides access to the dictionary attributes for this entry (starting with the Eureka release).
Labels	Provides access to the labels used for the table or field you are editing.

Default sort order

The system automatically sorts lists that meet one of the sort order criteria.

1. If there is a user preference specifying the sort order for the table, the system sorts by the field listed in the user preference.
2. If the table is the Task table or any of its extensions, the system sorts by the **Number** field.
3. If the table contains a field with the attribute **isOrder=true**, the system sorts by this field.
4. If the table contains an **Order** field, the system sorts by this field.
5. If the table contains a **Number** field, the system sorts by this field.
6. If the table contains a **Name** field, the system sorts by this field.

Enhancements

Fuji

- These changes support developing scoped applications:
  - For a new dictionary entry, the **Tables** field shows only the tables that that meet the scope protections for adding fields.
  - When administrators add a field to a table, the field name is prefixed according to the scope of the table and the scope of the field.

Eureka

- The Dictionary Entry form includes an Advanced view that provides additional fields than the default view provides. Not all fields on the Advanced view are new with the Eureka release.
- Administrators can configure reference fields using simple, advanced, or dynamic reference qualifiers.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/data-dictionary-tables/concept/c_SystemDictionary.html

Dictionary Attributes

Note: *This article applies to Fuji. For more current information, see Dictionary Attributes* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

Dictionary attributes alter the behavior of the table or element that the dictionary record describes. Administrators can add or modify dictionary attributes.

Adding an Attribute

To add an attribute to a table or field, navigate to the System Dictionary record for the Dictionary entry, and add the attribute to the **Attributes** field. Attributes are comma-separated; if attributes already exist on a dictionary record, add a comma, with no spaces, before adding a new attribute.

For an attribute that accepts *true/false* values:

- To specify a value of *true*, you can either enter *attribute* or *attribute=true*.
- To specify a value of *false*, you can either ensure that the attribute does not appear or enter *attribute=false*. To maintain values during upgrades, do not remove an attribute that is on a table by default. See Maintaining Attribute Values for Upgrades.

To specify a different attribute value for a field when it appears on extended tables than when it appears on base tables, see Dictionary Overrides.

Maintaining Attribute Values for Upgrades

If you remove an attribute that is part of the base system, it is automatically restored during an upgrade. To prevent upgrades from changing the desired behavior of your system, leave the attribute on the table or field, but set its value as desired.

For example, if a field has the attribute *knowledge_search=true* by default, do not remove the attribute to set it to false; rather set it to *knowledge_search=false*.

Available Attributes

This table lists the available attributes. The columns in this table are sortable.

Name	Value	Target element	Description
allow_null	true/false	field_name field	If present or true, allows entering "None" as the field
allow_references	true/false	field_name field	If true, a tree is displayed to select from that includes reference fields so you can dot-walk.
approval_user	name of field containing the user(s) for the approval type this field represents	integer field	The fields of the table are used to perform the lookup using a 'matcher'. Approvals are specified as fields in the table that have an attribute of 'approval_user=<field_name>', where <field_name> indicates the field in the table that contains the user(s) for this approval type. Fields with this attribute contain an integer value that indicates the sequence for the approvals. All approval fields with the same sequence number indicate that multiple approvals are required before continuing. Approvals are requested in the order of the sequence numbers. For example, all approvals with sequence number 100 must be approved before approvals with sequence number 200 are requested.
attachment_index	true/false	any table	If true, attachments on the table are indexed for search purposes. See Enabling Attachment Indexing.
base_table	name of base table type	table_name field	A table_name field allows the user to choose any table derived from the table specified by this attribute. By default, the base table itself is also included in the choice list (but see skip_root to turn off this behavior).
calendar_elements	list of field names, separated by semicolons (";")	any calendar event table	Specifies a list of fields to be used when constructing the description of a calendar event. If not specified, the usual display name plus short description are used. The the calendar_elements attribute does not support derived (dot-walked) fields.
collection_interval	interval specified as "HH:MM:SS" (like "01:02:30" for one hour, two minutes, and thirty seconds)	collection field	Specifies the interval of metrics collection.
close_states	inactive state integer values	task state field	Used by the TaskStateUtil API - identifies the list inactive state values delimited by semicolons (;)
critical	true/false	any field in the apm_application table	Defines fields that are critical information about an application. This allows tracking the entry of critical information.
disable_execute_now	true/false	any table derived from sys_auto	If present or true, disables the usual "Execute Now" button. This is used by applications using schedules (such as Discovery) to substitute their own more appropriate action.
default_rows	integer value	mulitext fields	Sets the default number of rows in a multitext field.

default_close_state	state integer value	task state field	Used by the TaskStateUtil API - identifies the default close state value for a task table
default_work_state	state integer value	task state field	Used by the TaskStateUtil API - identifies the default working state value for a task table
detail_row	name of field to display in detail row	any table	Displays the value of the specified field as a detail row for each record in the list view. This attribute is available starting with the Fuji release (UI14 or later required). **Note:** If different `detail_row` attributes are defined for a parent table and a child table, the system uses the child table attribute.
email_client	true/false	any table	If present or true, causes an icon (an envelope) to appear on the right side of the form's navigation bar, just to the left of the up and down arrows. If clicked, a popup email client appears.
extensions_only	true/false	any table	Table should only have records in tables that extend it. For example, the "task" table has this attribute because you would create incident, problem, change records and not task records.
field_list_selector	true/false	any glide_list	Allows the user to select a field from the dependent table (or current if dependent is not specified). This is used in some workflow activities.
field_decorations	UI Macro name list, separated by semicolons (";")	most fields (except multi-line text fields)	Similar to ref_contributions, causes the named UI Macro to be invoked when the field is rendered on a form.
format	format name	any numeric field such as integer or glide-date	Specifies the type of number format to use instead of the standard numeric formatting. Options are: • **glide_duration:** formats a time specified in milliseconds as *ddd hh:mm:ss*. • **none:** disables automatic number formatting where a comma character is inserted after every three digits (for example, changes 2,500 to 2500).
fv	table name; field name; sys_id	field_value field	Uses the three values to set the display of the field_value field.
glide.db.oracle.ps.query	true/false	any table	If present and false, prevents the use of Oracle prepared queries on the table.
global_visibility	true/false	any table with a sys_domain column	If present or true, makes this table visible globally even if there are domain restrictions (that is, the sys_domain field has a value).
hasLabels	true/false	any table	If present or true, marks this table as being the target of a label at some point. This attribute can be set manually, but it is set automatically whenever a label is generated. When true, the label engine will run on any change to the table, updating the labels as needed.
hasListeners	true/false	any table	If present or true, marks this table as available for listeners to get events (insert, update, delete) on.
hasWorkflow	true/false	any table	Tells the workflow engine to listen for changes to the table, firing events to a workflow when a record associated with a particular workflow has changed.
html_sanitize	true/false	any field	If present or true, HTML sanitization is enabled for the selected field.
icons	name of JavaScript class	any workflow field	Specifies a JavaScript class that produces workflow icons.

image	relative path of image file	any table	Specifies an image file to be used when the table is used in a module or BSM map. This specification overrides the icons that would otherwise be used for the table.
include_container_types	true/false	any internal_type field	Causes the field to render with container (split) types as well other types.
iterativeDelete	true/false	any table	If present or true, forces all row deletes to be executed iteratively. Otherwise, some deletes may be performed using a more efficient bulk method.
knowledge_custom	name of JavaScript function	any field	Specifies a JavaScript function to implement a custom knowledge search (see knowledge_search).
knowledge_search	true/false	string fields	If present or true, causes a knowledge search icon (a small book) to appear next to the field. Clicking this icon launches a pop-up window for searching the knowledge base, unless a custom knowledge search function has been specified (see knowledge_custom).
largeTable	true/false	any table	If present or true, marks this table as "large" for the purpose of preventing table locking with specific MySQL database operations (adding/removing a column/index, compacting a table). Without this attribute (or the smallTable attribute), whether a table is large is determined by the glide.db.large.threshold property, or the default value of 5,000.
listen	true/false	any field	If present or true, causes a call to a JavaScript function named <tableName>_<fieldName>Listen, or globalListen if that function does not exist. The function is called with arguments (tableName, fieldName, oldValue, newValue).
live_feed	true/false	any field	If present or true, creates a toggle option on the activity formatter header for incidents, tasks, and problems. The toggle provides the choice between between the **Live Feed** for that record (also known as a document feed) or the activity formatter fields already in use. See Activity Formatter for more details.
long_label	true/false	any field	Long or short labels refer to the label that is displayed for reference fields on a form. For example, if the field contains the caller's email address, the long label would be "Caller Email" while the short label would just be "Email." Usually the placement of the field on the form makes it clear what the field represents. The global property (glide.short.labels) is used to specify the type of labels that are displayed for all reference fields on any form. This global property can be overridden for any field by setting the short_label=true or long_label=true attribute for the field in the Dictionary.
mode_toggler	true/false	any composite_name field	If present or true, causes a name mode toggle icon (a small right-pointing triangle) to appear to the right of the label. Clicking this icon causes the field's rendering to change from a text field accepting <tablename>.<fieldname> to a pair of reference choice boxes (one for the table, the other for the field). The latter is the default.
model_class	binary Java class name	any field of type glide_var	Specifies a model variable within Java code. The model must have a class that implements the IVariablesModel interface.

model_field	see description	any field of type glide_var	Identifies a reference field in the record that has the model defined for it. For example, a workflow activity is associated with an activity definition. The activity definition has a related list of questions that make up the model for that activity definition. By using the activity_definition as the model_field for the activity, the model for the workflow activity is built by reading the questions that are defined for the referenced activity definition.
nibble_size	positive integer	any table affected by the table cleaner.	Specifies the maximum number of records the table cleaner can delete in a single operation. The default value for this attribute is 250.
nibble_sleep	true/false	any table affected by the table cleaner.	If false, causes the table cleaner to perform cleanup operations without a pause between each operation.
no_attachment	true/false	any table	If present or true, prevents the attachment icon (a paperclip) from appearing on the form's navigation bar (just to the left of the up and down arrows).
no_attachments	true/false	any table	If present or true, attachments will not be checked for and deleted when a record from this table is deleted. Meant for high-activity tables that never have attachments.
no_audit	true/false	any table	If present or true, this field will not be audited, even if the table is being audited.
no_audit_delete	true/false	any table	If present or true, a sys_audit_delete record will never be created when a record from this table is deleted. Meant for high-activity tables that never need sys_audit_delete information.
no_auto_map	true/false	any table	If true, this field will not be mapped during an import set. This is primarily used for LDAP imports.
no_email	true/false	any glide_list field referencing sys_user	If present or true, the email box is removed from the glide_list field like the "Watch list" field.
no_multiple	true/false	any glide_list field	Hides the select multiple icon.
no_optimize	true/false	any table affected by the table cleaner.	If present or true, prevents the MySQL table compaction operation from running on the specified table. The table compaction operation normally runs after the table cleaner deletes at least 50% of the data in the specified table.
no_separation	true/false	any table	If present or true, marks this table as not participating in domain separation.
no_text_index	true/false	any field on a text indexed table	If a table is text indexed, the no_text_index attribute on a field will prevent this field from being included in the text index.
no_truncate	true/false	any string field	In a list view, shows the entire text value of the multi-text value in a list, without truncating it. Without this attribute the string is truncated based on the UI Property 'Number of characters displayed in list cells' which is 40 by default.
no_update	true/false	table	Is *true* for tables in which records are inserted or deleted but not updated. Prevents the system from creating sys_mod_count, sys_updated_by, sys_updated_on fields in the table when it is created. Does not stop the table from being updated. This attribute is used to save space on high volume system tables, such as syslog and sys_audit.
no_view	true/false	any glide_list field	Hides the view selected item icon.

onlineAlter	true/false	any table	Tables with the onlineAlter attribute perform MySQL database operations using online schema changes. Online schema changes provides a lock-free table upgrade when adding, modifying, or removing columns and when adding or dropping indexes. Without online schema changes, these changes to the database lock write access during execution. Online schema changes use additional system resources. The onlineAlter attribute is enabled by default for all tables. Oracle databases do not lock tables by default and do not use online schema changes.
order	numeric value	model variable fields	Used internally only (for model variables).
popup_processor	binary Java class name	any field or table	Specifies a custom popup processor for processing the field (or all fields in a table).
readable	true/false	any conditions field	When true, causes the conditons field to be rendered in any list view as a human-readable condition (instead of the encoded query actually stored in the database). The form view for this field is unaffected.
ref_ac_columns	list of field names separated by semi-colons	any reference field with an auto completer (see ref_auto_completer)	Specifies the columns whose display values should appear in an auto completion list in addition to the name. See the cmdb_ci field ("Configuration Item") on the Incident form for a working example.
ref_ac_columns_search	true/false	any reference field with an auto completer (see ref_auto_completer)	Causes auto-complete to work with all fields specified in the ref_ac_columns attribute. This overrides the default behavior, which searches only the display value column. See Configuring Auto-Complete for Any Field in the Reference Field List.
ref_ac_display_value	true/false	any reference field with an auto completer (see ref_auto_completer)	Causes the reference field to hide a the display value column so that auto-complete only matches text from the columns listed in the ref_ac_columns attribute. This feature requires the use of the AJAXTableCompleter class and the **ref_ac_columns**, **ref_ac_columns_search**, and **ref_ac_display_value attributes**. See Removing the Display Value Column.
ref_ac_order_by	field name	any reference field with an auto completer (see ref_auto_completer)	Specifies the column that will be used to order the auto completion list.
ref_auto_completer	JavaScript class name	any reference field (can be applied to a table to affect all reference fields on the table.)	Specifies the name of a JavaScript class (client side) that creates the drop-down auto completion choices. Valid class values include: • **AJAXReferenceCompleter**: Displays matching auto-complete choices as a drop-down choice-list. The list only displays the reference table's display value column. Reference fields automatically use this class if there is no other auto-completion class specified. • **AJAXTableCompleter**: Displays matching auto-complete choices as rows in a table. The table displays the reference table's display value column and any columns listed in the **ref_ac_columns** attribute. • [**AJAXReferenceChoice**: Displays matching auto-complete choices as a drop-down choice-list. The list only displays the reference table's display value column. Furthermore, the list only displays up to 25 matching choices. If there are more than 25 auto-complete choices, the reference field instead displays the choices with the AJAXTableCompleter class. For more information, see Auto-Complete for Reference Fields.
ref_contributions	UI Macro name list, separated by semicolons (";")	any reference field	Causes the named UI Macro to be invoked when the field is rendered on a form.

ref_list_label	label text	any table	Specifies the title to use in a list banner.
ref_qual_elements	field name list, separated by semicolons (";")	any reference field with a reference_qual field	Specifies a list of fields to be sent back to the server in order to get an updated reference.
ref_sequence	list of fields in referenced table, separated by top hats ("^")	any reference field	Specifies the fields in the referenced table that should be used to order the choice list. This works like an ORDER BY clause in SQL, with each element in ascending order.
reference_types	list of valid reference types that are clickable separated by semicolons (";")	field_name field	Limits the reference fields that are displayed in the tree to the specified types.
remoteDependent	name of database and table (like "model.matcher")	any script field	Defines the remote (such as, in another database) table that the script depends on.
repeat_type_field	field name	a repeat count field for schedule rotation	Specifies the field that contains the repeat type (daily, weekly, monthly, or yearly).
restrictTo	field name (including indirect, dot-walked field references)	any conditions field	Specifies the field that contains the comma-separated list of fields that the conditions should be restricted to using.
ro_collapsible	true/false	any multi-line field	If present or true, causes an icon (either a "+" or a "-") to appear next to the field's label, allowing the field itself to be expanded or collapsed.
scale	integer	decimal field	Sets the number of decimal places to use on the Decimal field type. The default is 2. This is applied to the Max Length of the field. **Note: Increase the Max Length to a value greater than 15 to increase this attribute.**
script	a function that returns the contents of the field	any slushbucket field	Allows you to write a script to define what will be loaded into the slushbucket field.
short_label	true/false	any field	Long or short labels refer to the label that is displayed for reference fields on a form. For example, if the field contains the caller's email address, the long label would be "Caller Email" while the short label would just be "Email." Usually the placement of the field on the form makes it clear what the field represents. The global property (glide.short.labels) is used to specify the type of labels that are displayed for all reference fields on any form. This global property can be overridden for any field by setting the short_label=true or long_label=true attribute for the field in the Dictionary.
show_all_tables	true/false	document ID fields	Allows users to select documents from system tables. For example, sys_script or sys_user. By default, users cannot select records from system tables.
show_condition_count	true/false	condition fields	Enables or disables the Condition Count Widget to preview how many records would be returned by a set of conditions.
skip_root	true/false	table_name field	If present or true, removes the base table from the choice list (see base_table for more details).
sla_basis	list of table names separated by semicolons (";")	any field of date type (glide_date_time, glide_date, due_date, date, or datetime)	Defines the tables for which this field determines the start (open) time of an SLA.

sla_closure	list of table names separated by semicolons (";")	any field of date type (glide_date_time, glide_date, due_date, date, or datetime)	Defines the tables for which this field determines the end (close) time of an SLA.
slushbucket_ref_no_expand	true/false	any reference field	If present or true, prevents users from expanding the field from a form or list slushbucket.
smallTable	true/false	any table	If present or true, marks this table as "small" (that is, not large) for the purposes of our querying strategy. Without this attribute (or the largeTable attribute), whether a table is large is determined by the glide.db.large.threshold property, or the default value of 5,000.
start_locked	true/false	any glide_list field	Determines whether the field is locked or unlocked by default. Set the value to **false** to unlock the field by default.
staticDependent	name of table	any script field	Defines the table that the script depends on.
strip_html_in_pdf	true/false	any field	Attempts to remove HTML tags from a field when that field is exported to a PDF. Most likely useful on HTML Fields.
synch_attachments	true/false	any table	Similar to update_synch but writes the record's file attachments to update sets. See click here to learn more.
table	name of table	field_name field	Displays the fields of the table specified.
tableChoicesScript	name of script include	table_name field	The name of a script include whose **process()** method returns an array of table names from which to select.
target_form	name of form	any table	Specifies the alternative form to be used when this table is referenced through a popup on a reference field.
text_index_filter_junk	true/false	any table	Set the value to **false** to disable the junk filter for the table. By default, Zing does not index or search for 2-digit numbers and single character words (unless they are Chinese or Japanese characters). You must regenerate the index after disabling the junk filter. This attribute results in a larger table index. For optimal performance, do not apply it unless it is required.
text_index_translation	true/false	any table	If present or true, forces indexes to be recalculated when translated strings are added. Default is not present.
text_search_only	true/false	table_name field	Limits the tables listed to those that are searchable by text.
time_zone_field	name of field containing the time zone	any date or date/time field	Specifies the field in the parent record that contains the reference time zone for this field.
timeDimension	true/false	any field of date type (glide_date_time, glide_date, due_date, date, or datetime) in a table subclassed from the task table	If present or true, enables production of time dimension data for use by OLAP (to produce reports based on quarters, weeks, or other time periods). **Note:** OLAP functionality has been deprecated. For reports based on time dimensions, see Trend Charts, Control Charts, or Trendbox Charts.
tree_picker	true/false	reference field with reference to a hierarchical table	Displays the hierarchy of reference values in a tree display (such as locations).
ts_weight	integer value	any field	Controls the relative importance of a match in the field for text search. See Controlling Match Relevance by Field.
types	list of valid element types separated by semicolons (";")	field_name field	Limits the fields display to the specified types.

update_exempt	true/false	field on any table where update_synch=true	If present or true, you can change this field without skipping updates to the rest of the record. During software upgrades, the value of this field is preserved, while the rest of the record receives upgrades. By default, the **Active** field on a tracked table is treated as update_exempt even if the attribute is not present. See Excluding Fields from Updates.
update_synch	true/false	any table	Indicates that changes in the table are tracked in update sets. Administrators cannot modify this attribute. To migrate data, use an instance-to-instance import.
use_workflow	true/false	any table that has delivery plans or uses workflow	If present or true, causes workflow to be used instead of delivery plans.
user_preference	true/false	any field	If present or true, causes any user preferences to be used instead of the normal default value.

Enhancements

Fuji

- The `detail_row` dictionary attribute is available to add a detail row to records in a list.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/reference-pages/concept/c_DictionaryAttributes.html

Dictionary Overrides

Note: *This article applies to Fuji. For more current information, see Dictionary Overrides* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

Dictionary overrides [2] provide the ability to define a field on an extended table differently from the field on the parent table. For example, for a field on the Task [task] table, a dictionary override can change the default value on the Incident [incident] table without affecting the default value on Task [task] or on Change [change].

Administrators can override these aspects of a field:

- Reference qualifiers
- Dictionary attributes
- Default values
- Calculations
- Field dependencies
- Default column display values
- Mandatory and read-only status

Video Tutorial

This video demonstrates how to use dictionary overrides through an example of changing the initial State value for incidents and problems to differ from that of their parent table.

ServiceNow Dictionary Override Best Practices

Defining Dictionary Overrides

Define dictionary overrides from a related list on the dictionary entry form for the field. You can add dictionary overrides only on tables that are in the same scope as the dictionary override (starting with the Fuji release).

To define a dictionary override:

1. Navigate to **System Definition > Dictionary**.
2. Open the record for the field.
3. In the **Dictionary Overrides** related list, click **New**.
4. Fill in the fields on the form, as appropriate (see table).
5. Click **Submit**.

Field	Description
Table	Select the extended table to which the dictionary override applies.  **Note:** *The list shows only tables and database views that are in the same scope as the dictionary override (starting with the Fuji release).*
Override reference qualifier	Select the check box to display the **Reference qualifier** field, which overrides the reference qualifier for the field on the extended table.
Override dependent	Select the check box to display the **Dependent** field, which overrides the field on which the current field depends.
Override attributes	Select the check box to display the **Attributes** field, which overrides the dictionary attributes for the field on the extended table. **Note:** Any attributes defined on the base table are ignored. If there are attributes on the base table that should still apply to the extended table, make sure to include them in this field.
Override default value	Select the check box to display the **Default Value** field, which overrides the default value for the field on the extended table.
Override calculation	Select the check box to display the **Calculation** field, which overrides the calculation of the value for the field on the extended table.
Override mandatory	Select the check box to display the **Mandatory** field, which overrides whether the field on the extended table must contain a value to save a record (starting with the Dublin release).
Override read only	Select the check box to display the **Read only** field, which overrides whether a user can change the field value on the extended table (starting with the Dublin release).
Override display value	Select the check box to use this field as the display value on the extended table. For example, the Story [rm_story] table uses the short description as the display value in reference fields instead of the number, as defined in the Task [task] table.

Enhancements

Fuji

- These changes support developing scoped applications:
  - The **Table** field shows only tables that are in the same scope as the dictionary override.
  - The **Application** field is added on the form view.

Dublin

- Administrators can override the mandatory and read-only status of fields on extended tables.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/data-dictionary-tables/concept/c_DictionaryOverrides.html

[2] https://docs.servicenow.com/bundle/helsinki-servicenow-platform/page/administer/data-dictionary-tables/concept/c_DictionaryOverrides.html

Data Management and Archiving

Archiving Data

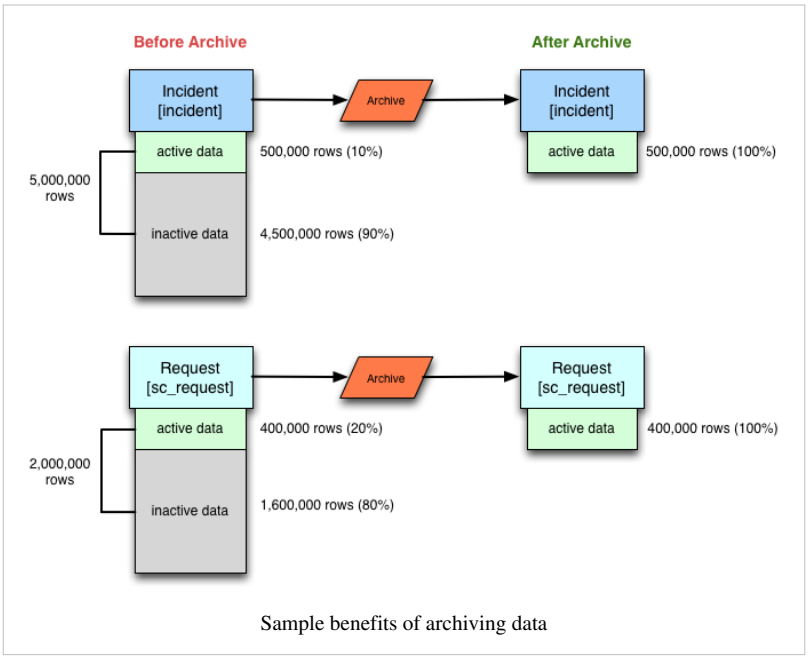
Note: *This article applies to Fuji. For more current information, see Data Archiving* [1] *at* http://docs.servicenow.com The Wiki page is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

The longer an instance runs, the more likely it is to accumulate data that is no longer relevant to current business needs. For example, task records from two years ago are typically less relevant than currently active tasks. Old data may eventually cause performance issues by consuming system resources and slowing down queries and reports.

If you cannot delete this data because you need it for auditing or for historical purposes, archive the data to remove it from immediate access and free up system resources. The archive application moves data that is no longer necessary for immediate day-to-day access from primary tables into a set of archive tables.

Data archiving supports domain separation. For example, incidents that belong to a domain keep their domain designation even after they are archived.



Sample benefits of archiving data

Data Archiving Setup

To setup data archiving:

1. Activate the **Data Archiving** plugin.
2. Create an archive rule for the table you want to archive.
3. Define how the archive rule handles related records.
4. Verify how many records the archive rule affects.
5. Activate the archive rule.

Activating the Data Archiving Plugin

Click the plus to expand instructions for activating a plugin.

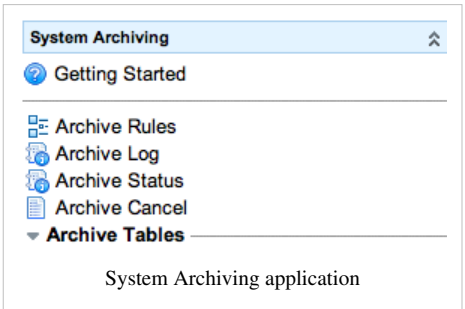
If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

The **Data Archiving** plugin installs the System Archiving application.



System Archiving application

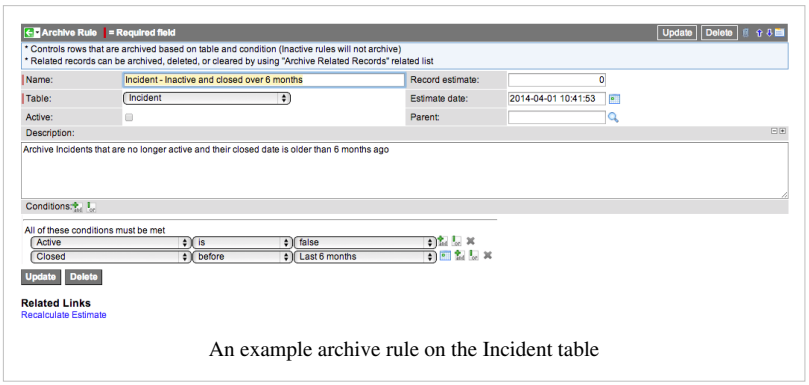
Creating an Archive Rule

The System Archiving application includes several sample archive rules that illustrate the archive features. The sample archive rules are not intended as best practice and are inactive by default.

To create your own archive rules:

1. Navigate to **System Archiving > Archive Rules**.
2. Click **New**.
3. Fill in the fields as appropriate (see table).
4. Click **Submit**.

Note: *ServiceNow recommends leaving your archive rules inactive until you calculate an estimate of the number of records the rule affects and verify that rule behaves as expected.*



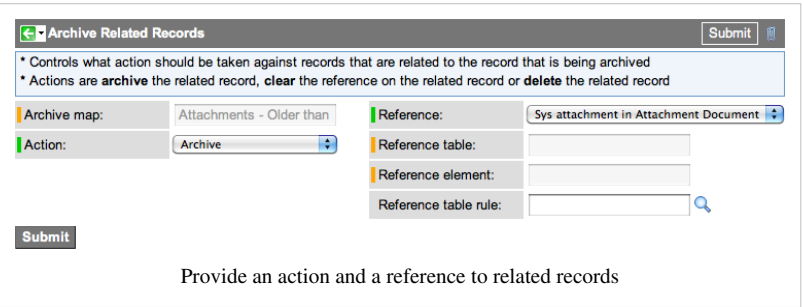
An example archive rule on the Incident table

Field	Description
Name	*Required.* Enter a unique name that identifies the rule. Since this is the display field for archive rules, references to archive rules display the name you enter here.
Table	*Required.* Select the table containing records to archive. There can only be one archive rule per table. If there is an existing archive rule for a table, the table no longer appears as an option on the archive rule table list.
Active	Select this check box to enable the archive rule. Clear the check box to disable the rule. ServiceNow recommends leaving your archive rules inactive until you calculate an estimate of the number of records the rule affects and verify that rule behaves as expected.
Description	Enter a description of the archive rule.
Conditions	*Required.* Select the fields and values that must be true in order for the archive rule to run. Typically, you would archive inactive records older than a certain date.
Parent	Select an existing archive rule on which this rule depends to run. This archive rule only runs when the parent rule also runs.

Archive Related Records

Use the **Archive Related Records** related list to add related records to the archive rule.

1. Navigate to **System Archiving > Archive Rules**.
2. Select the appropriate archive rule.
3. From the **Archive Related Records** related list, click **New**.
4. Fill in the Archive Related Records form. See the table for field descriptions.
5. Click **Submit**.



Provide an action and a reference to related records

The Archive Related Records form uses these fields.

Field	Description
Archive map	Displays the archive rule to which the related records apply.
Action	Select the action you want the archive rule to take on related records. Choices include: • **Archive:** archive records that reference the archived record. • **Clear:** remove the reference to the archived record. The record no longer references the archived record and does not appear as a related record in future archives. • **Delete:** delete records that reference the archived record.
Reference	Select the relationship of the records you want to apply an action to. The **Reference** field lists all relations that currently exist for the table being referenced. There are two types of possible relations.

- Another table has a reference field calling the archived table. For example, if you are archiving problem records, there is a **Problem ID** field in the Incident table that references the related problem records.
  - The **Archive** action archives the related record in addition to the primary record. For example, if you select the **Problem in Incident** reference, the related record rule also archives any incident record that references an archived problem.
  - The **Clear** action removes the reference to the primary record. For example, if you select the **Problem in Incident** reference, the related record rule updates any incident record with a reference to the archived problem record by clearing the reference. If the reference is a many-to-many relationship, the related record rule deletes the reference instead of clearing the reference.
  - The **Delete** action deletes any record that references the primary record. For example, if you select the **Problem in Incident** reference, the related record rule deletes any incident record that references the archived problem record.
- Another table has a **Document ID** field which might point to the archived table. For example, if you are archiving problem records, there is a **sys ID** field in the Attachments table that may reference the problem record. The list indicates document ID relationships by displaying an asterisk (*) character at the end of the selection name.
  - The **Archive** action updates the **Document ID** of the related record to point to the archived table. For example, if you select the **Table sys ID Attachment(sys_attachment)*** reference, the related record rule updates the attachment record to change the **Document ID** to refer to the archived table record.
  - The **Clear** action updates the **Document ID** of the related record to point to the archived table.
  - The **Delete** action deletes any record that references the primary record. For example, if you select the **Table sys ID Attachment(sys_attachment)*** reference, the related record rule deletes any attachment record that references the archived primary record.

Reference table	Displays the table where the rule looks for related records.
Reference element	Displays the reference field or **Document ID** the rule queries for.
Reference table rule	Select an existing archive rule that applies to the related records you are archiving. For example, if you already have an archive rule for the Incident table, you can select the existing Incident table rule when archiving records related to incidents records.

Verify the Number of Records Affected

Each archive rule provides an estimate of the number of records the rule affects in the **Record estimate** field. This estimate only includes primary records and excludes any related records added to the archive rule. The estimate helps you determine if the archive rule affects the number of records you expect it to. If the estimate is too high or low, change the archive rule conditions and then click the **Recalculate Estimate** related link.

1. Navigate to **System Archiving > Archive Rules**.
2. Select the archive rule you want to estimate records for.
3. Click the **Recalculate Estimate** related link.

Activating the Archive Rule

After verifying that the archive rule is selecting records as expected, activate the archive rule. The instance runs the archive rule as a schedule job set to run every 60 minutes.

1. Navigate to **System Archiving > Archive Rules**.
2. Select the archive rule you want to activate.
3. Select the **Active** check box.
4. Click **Update**.

Activating an Archive Rule Immediately

If you do not want to wait for the scheduled job to run the archive rule, you can manually start the archive rule.

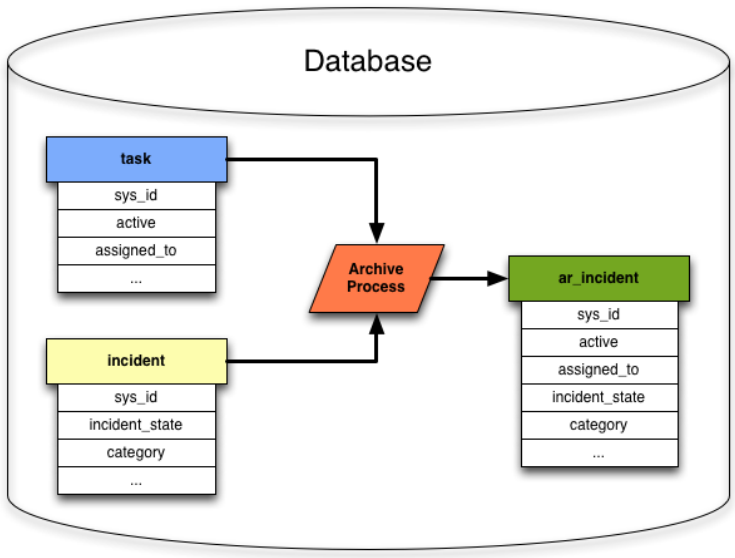
1. Navigate to **System Archiving > Archive Rules**.
2. Select the archive rule you want to run.
3. Click the **Run Archive Now** related link.

Note: *The archive rule must be active in order to see the link.*

Tables and Modules Created by an Archive Rule

The first time you activate an archive rule, it performs the following actions:

- Creates the archive table in the database. The archive table has the same name as the primary table with an “ar_” prefix. For example, if you archive the Incident [incident] table, then the archive table is [ar_incident].
- Converts multiple joined tables into a single flat-file archive table. The archive table no longer consists of a base and extended tables.
- Converts reference field values (values set by references to records in other tables) into string values. The archive record contains the display value of the reference field at the time of the archive.
- Adds a module to the **Archive Tables** list in the **System Archiving** application. The module name is a combination of the word "Archive" plus the display name for the archived table. For example, the archive module for the Attachment [sys_attachments] table is **Archive Attachment**. Click the module name to view records in the archive table.
- Creates a list of the archive table using the default list view.
- Creates a form for the archive table using the default form view. The form excludes any dot-walking fields such as **Caller ID.Email**.
- Creates an activity formatter and variable formatter if they exist for the table being archived.



Conversion of multiple joined tables into a single flat archive table

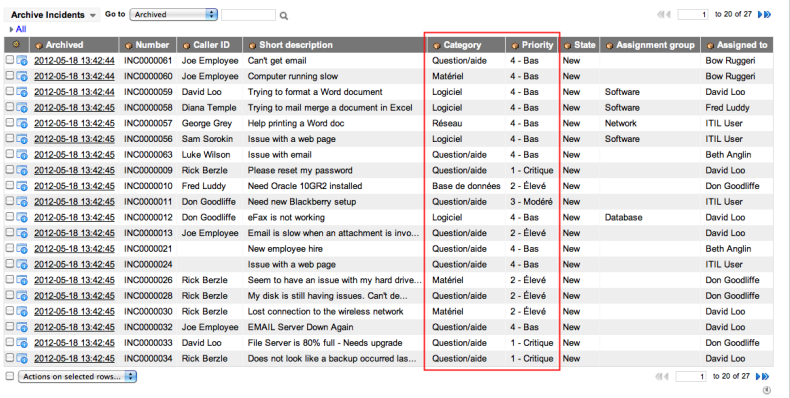
Reference Values Converted to Strings

Archived data is stored as a flat file with no reference fields to other tables. The archive process converts any references to other tables to string values. In the case of a reference field, the string uses the display value such as the caller's user name. For example, the **Caller** reference field in an incident would display the string **ITIL User**. If the reference was a document ID and the archive rule included the option to archive related document IDs, then the string is the document ID of the related record.

It is important to note that archive records do not receive any future changes to referenced values. For example, if you change the user name for “John Smith” to “John A Smith”, all active incident records automatically show the caller as “John A Smith” because of the reference between the Incident and User tables. However, all archived incident records display the user name that existed at the time of the archive. Any incident for “John Smith” continues referencing this user. Likewise, if you delete a user from the system, current incidents no longer display the deleted user as a caller. However, there can be archived incidents that still display the string “John Smith” as the user because the user existed at the time of the archive.

Setting the Language of Archived Strings

On internationalized instances, the archive process uses the language of the SYSTEM user to select the display value strings. If there is no SYSTEM user, the instance uses the default language setting to select the display value strings. You can either create a SYSTEM user with a specific language setting or set the system default language to select the language of archived strings.


Archive Incidents Go to Archived 1 to 20 of 27

All

	Archived	Number	Caller ID	Short description	Category	Priority	State	Assignment group	Assigned to
	2012-05-18 13:42:44	INC0000061	Joe Employee	Can't get email	Question/aide	4 - Bas	New		Bow Ruggeri
	2012-05-18 13:42:44	INC0000060	Joe Employee	Computer running slow	Matériel	4 - Bas	New		Bow Ruggeri
	2012-05-18 13:42:44	INC0000059	David Loo	Trying to format a Word document	Logiciel	4 - Bas	New	Software	David Loo
	2012-05-18 13:42:45	INC0000058	Diana Temple	Trying to mail merge a document in Excel	Logiciel	4 - Bas	New	Software	Fred Luddy
	2012-05-18 13:42:45	INC0000057	George Grey	Help printing a Word doc	Réseau	4 - Bas	New	Network	ITIL User
	2012-05-18 13:42:45	INC0000056	Sam Sorokin	Issue with a web page	Logiciel	4 - Bas	New	Software	ITIL User
	2012-05-18 13:42:45	INC0000063	Luke Wilson	Issue with email	Question/aide	4 - Bas	New		Beth Anglin
	2012-05-18 13:42:45	INC0000009	Rick Berzle	Please reset my password	Question/aide	1 - Critique	New		David Loo
	2012-05-18 13:42:45	INC0000010	Fred Luddy	Need Oracle 10GR2 installed	Base de données	2 - Élevé	New		Don Goodliffe
	2012-05-18 13:42:45	INC0000011	Don Goodliffe	Need new Blackberry setup	Question/aide	3 - Modéré	New		ITIL User
	2012-05-18 13:42:45	INC0000012	Don Goodliffe	eFax is not working	Logiciel	4 - Bas	New	Database	David Loo
	2012-05-18 13:42:45	INC0000013	Joe Employee	Email is slow when an attachment is invo...	Question/aide	2 - Élevé	New		David Loo
	2012-05-18 13:42:45	INC0000021		New employee hire	Question/aide	4 - Bas	New		Beth Anglin
	2012-05-18 13:42:45	INC0000024		Issue with a web page	Question/aide	4 - Bas	New		ITIL User
	2012-05-18 13:42:45	INC0000026	Rick Berzle	Seem to have an issue with my hard drive...	Matériel	2 - Élevé	New		Don Goodliffe
	2012-05-18 13:42:45	INC0000028	Rick Berzle	My disk is still having issues. Can't de...	Question/aide	2 - Élevé	New		Don Goodliffe
	2012-05-18 13:42:45	INC0000030	Rick Berzle	Lost connection to the wireless network	Matériel	2 - Élevé	New		David Loo
	2012-05-18 13:42:45	INC0000032	Joe Employee	EMAIL Server Down Again	Question/aide	4 - Bas	New		David Loo
	2012-05-18 13:42:45	INC0000033	David Loo	File Server is 80% full - Needs upgrade	Question/aide	1 - Critique	New		Don Goodliffe
	2012-05-18 13:42:45	INC0000034	Rick Berzle	Does not look like a backup occurred las...	Question/aide	1 - Critique	New		David Loo

Actions on selected rows... 1 to 20 of 27

Sample archive with display value strings set to French

Setting Archive Rule Processing Behavior

In order to prevent the archive process from consuming too many system resources, the instance uses several system properties to control how many records the archive rule processes at one interval. By default an archive rule follows these processing rules:

- Archives 100 records for each batch job
- Sleeps 1 second between batch jobs
- Runs 10 batch jobs in an archive run (every hour)

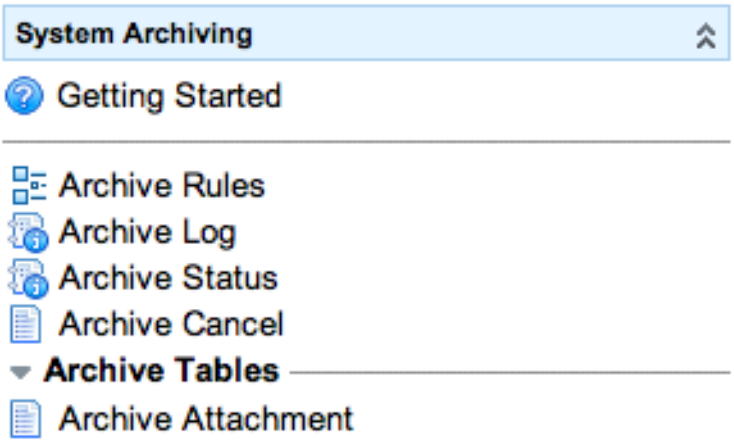
You can change these default settings by adding the following system properties.

Name	Description	Type	Default Value
glide.db.archive.batch_size	Controls how many records an archive rule processes per batch job.	Integer	100
glide.db.archive.sleep_time	Controls how long each archive rule batch job runs, in seconds.	Integer	1
glide.db.archive.max_iterations	Controls the maximum number of batch jobs to run within an hour.	Integer	10

Viewing Archived Data

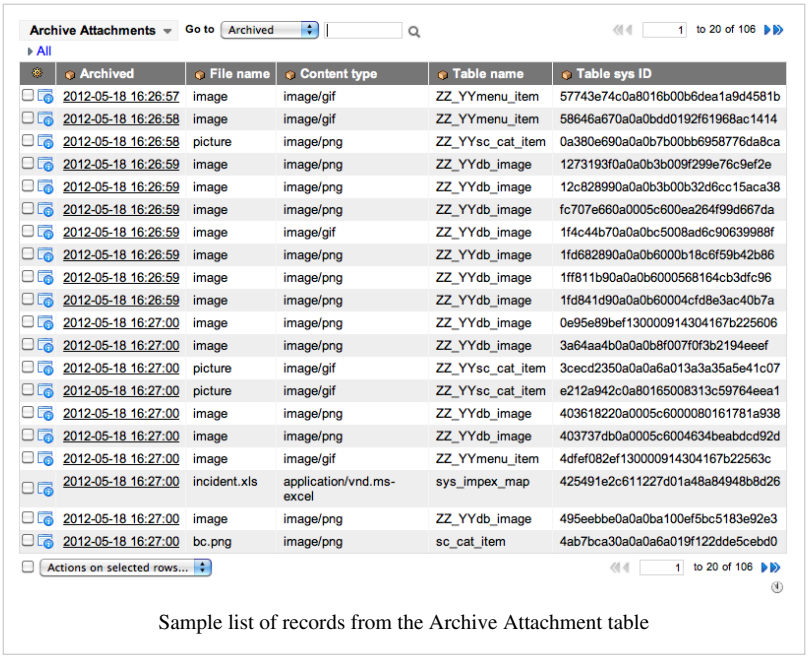
You can view archived tables from the **Archived Tables** list.

1. Navigate to **System Archiving > Archive Tables**.
2. Click the module name for the archived table you want to view. For example, **Archive Attachment**.



Sample module created for Archive Attachment table

3. Click a record from the archived table to see the archived record.



Sample list of records from the Archive Attachment table

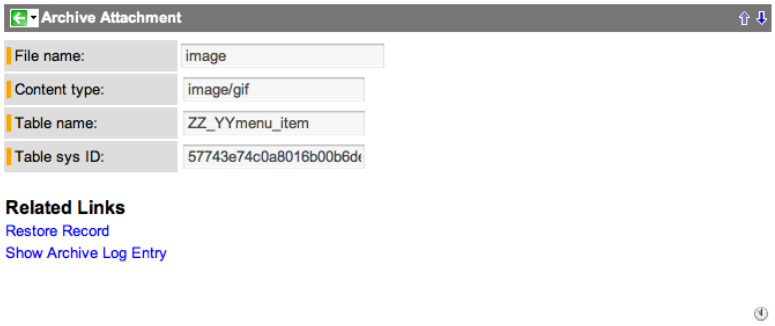
Querying Archived Tables

Archived tables are not optimized for ad hoc queries. They only contain index entries for the display value, creation date, and the primary key of sys_id. For this reason ServiceNow does not recommend making ad hoc queries against an archived table, such as searching for all priority 1 archived incidents. Instead, only search against the indexed fields. For example, search for incident INC100001 or incidents created on a specific date.

Restoring Archived Data

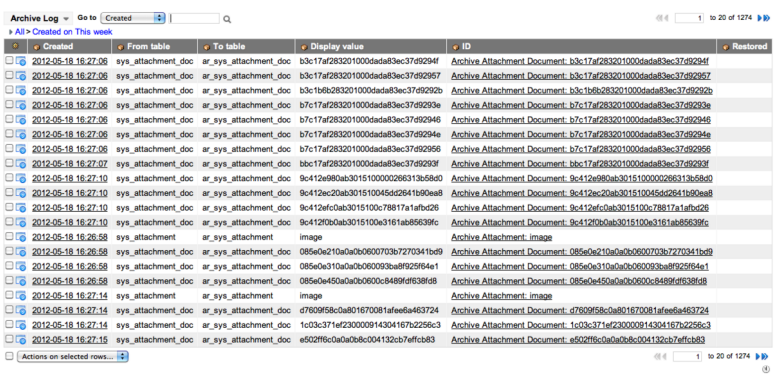
When you restore a record, the instance inserts it back into the primary table and flags the record as having been restored in the log.

1. Navigate to **System Archiving > Archive Log**.



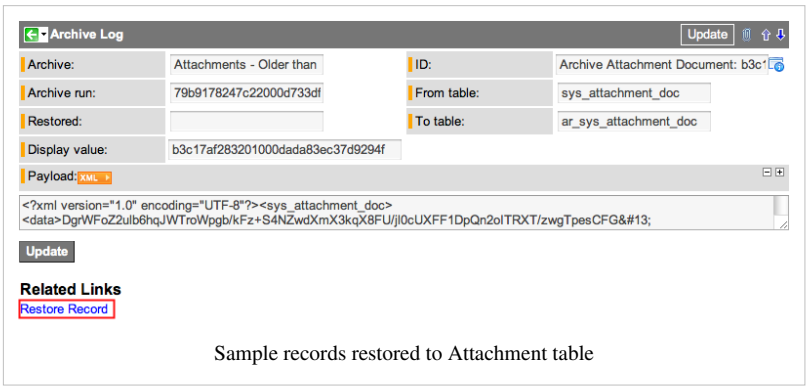
Sample record from the Archive Attachment table

2. Select the archived record to restore.



Sample archive log

3. Click the **Restore Record** related link.



Sample records restored to Attachment table

	Warning: Do not delete archive record log entries. Deleting an archive log entry prevents you from restoring the data for the archived records.

Archiving Restored Records

Archive rules will not archive restored records. In order to archive a restored record you must manually archive the record.

1. Navigate to the restored record.
2. Click the **Archive Record** link.

Each restored record has a related link to archive the record again

Archive Schedule

All active archive rules are executed by a system scheduled job set to run every 60 minutes. You can modify the job if you need to change the interval.

1. Navigate to **System Scheduler > Scheduled Jobs**.
2. Open the **Archive** record.
3. Modify the **repeat** value.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/database-rotation/concept/c_ArchiveData.html

Database Rotation

Overview

This page describes the approaches to managing table size growth and archiving old data. With data constantly being added to the system, and activity being logged into system tables in the database, these tables grow in size and require management. As data sets increase in size, the amount of I/O traffic associated with actions such as cleaning, deleting, and archiving can negatively effect the performance of an instance. Additionally, working with all rows in a data set, rather than a smaller working set, can create unnecessary risk.

The Database Rotation plugin preserves instance performance and averts risk associated with querying growing data sets utilizing two techniques. Both techniques are based on the concept of managing large quantities of data by separating whole sets into individual tables based on user-specified time parameters. After this task is performed, each technique handles data in a different manner:

- *Table Rotation* works by rotating among a small set of tables, and deleting and reusing the old tables for new data.
- *Table Extension* works by periodically starting a new table and allowing old tables to be easily archived and removed from the system.

For an additional archiving option see Archiving Data.

Usage

Table Rotation can be used for tables that:

- Hold data that is only valuable for a limited period of time
- Are written sequentially by inserting new rows

Table Extension can be used for tables:

- That hold data that's organized by time
- Where data is written and then never updated, or where updates are mostly focused on recent data
- Where data is needed for a period of time (e.g., two years) and can subsequently be discarded or archived off the system

Included with this Plugin

The Data Management and Archiving techniques are contained in two separate plugins:

- **Database Rotations Plugin** activates Table Rotation and Extension without any tables automatically included (com.snc.db.rotation)
- **Database Rotations Default Tables Plugin** applies Table Rotation and Extension to specific tables (com.snc.db.rotation_default_tables)

Functionality	Tables applied to
Table Rotation	syslog sys_querystat ecc_queue ecc_event cmdb_metric sysevent
Table Extension	sys_audit sys_email

Activating the Plugin

	Note: Deployment of this plugin should be executed in partnership with a ServiceNow representative.

For new instances, both plugins are activated by default.

For an existing instance without the plugins, activate the **Database Rotations** plugin:

Click the plus to expand instructions for activating a plugin.

If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

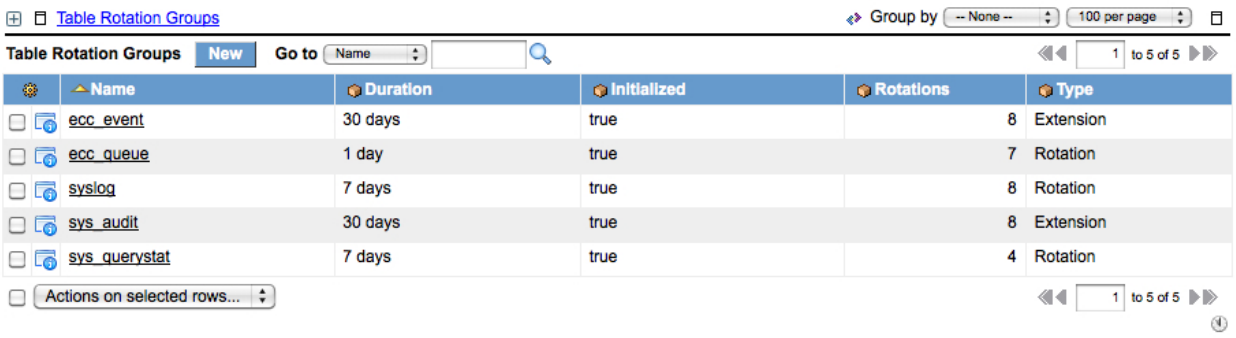
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Do not activate the **Database Rotations Default Tables** plugin. Instead, specify the tables manually, after consulting a ServiceNow representative.

Features

The System Definition Table Rotation module allows you to define a new table rotation, a new table extension or modify an existing one.



- **Name**: auto-generated from table name
- **Duration**: overall time parameter for function

- **Initialized**: sets function as active (true) or inactive (false)
- **Rotations**: number of tables to be created within Duration
- **Type**: indicates Extension (archiving) or Rotation (deletion) functionality

When you define a new rotation, a schedule is created and new data is subsequently written to one of the tables in the rotation group. You'll notice the that the group includes the original table plus a number of additional tables. Be aware that deleting a rotation will delete the additional tables and all the data, therefore the rotation should not be deleted if the data is needed.

Implementation

See individual pages to implement either Table Rotation or Table Extension.

Table Rotation

Note: *This article applies to Fuji and earlier releases. For more current information, see Table Rotation* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

Note: Deployment of this plugin should be executed in partnership with a ServiceNow representative.

Overview

Functionality

Table Rotation preserves instance performance and averts risk associated with querying growing data sets by using the External Communication Channel (ECC) Queue and the **sys_created_on** field to separate data sets into multiple tables based on date. The administrator specifies the time parameter (Duration) of the process and the number of tables (Rotations) within. After the rotation writes the last table in a rotation, the rotation overwrites the first table in the rotation. Please contact ServiceNow Technical Support before applying table rotation to a custom table. For example:

- The query 'Records created between 2015/12/10 08:49 and 2015/12/09 07/34 where topic=SystemCommand' is translated to a SQL query on a single table, because the clause on sys_created_on targets a single shard.
- The query 'Records updated between 2015/12/10 08:49 and 2015/12/09 07/34 where topic=SystemCommand', or without a date range, needs to target all shards and therefore is translated as a union query on all shards.

Advantages to Table Rotation

- Allows deletion of old data without affecting current data (for example, to drop or truncate a table).
- Ensures tables only grow to a reasonable size.
- Reduces working set of data when date is known for query.

Disadvantages

- Queries that do not use the table rotation date (for example, by using the **sys_updated_on** field), force an inefficient union query to query time ranges that span multiple tables and can be extremely slow if the number of sub-tables is large. To improve performance, it is recommended that the query includes a window of created dates.

Use

Table Rotation can be used for:

- Sequentially-written tables
- Insert-only tables

Table Rotation cannot be used for:

- sys_import tables
- Tables that extend the Task table

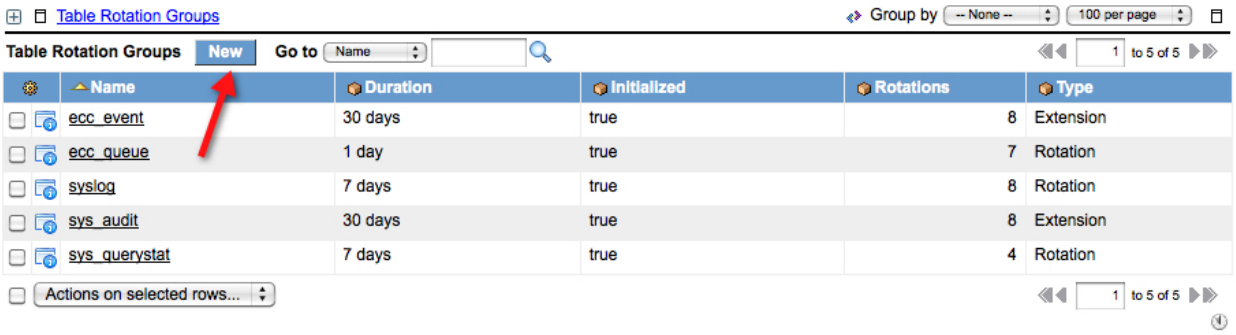
Example

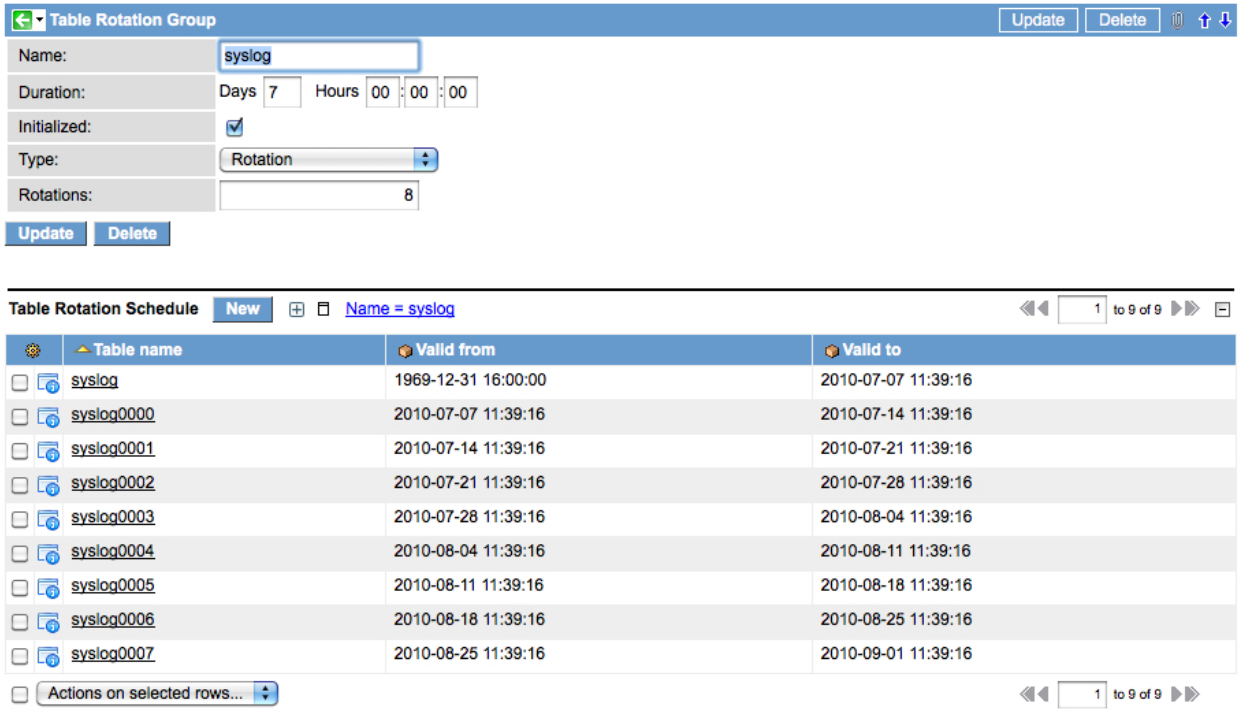
To apply Table Rotation:

1. Navigate to **System Definition > Table Rotation**.
2. Click **New**.
3. Set the rotation fields.
4. Click **Update**.

When you define a new rotation, a schedule is created and new data is subsequently written to one of the tables in the rotation group. You'll notice that the group includes the original table plus a number of additional tables.

Note: *Deleting a rotation will delete the additional tables and all the data. Do not delete the rotation if the data is needed.*





Rotation Fields

Field	Description
Name	Enter the name of the table to which the action is to be applied. In this case, Table Rotation to table *syslog*.
Duration	Set the overall duration for the action. 7 days is the duration in this example.
Initialized	Sets table rotation as active (true) or inactive (false). [This must be checked (true) for the rotation or extension to be active.]
Type	Choose either Table Rotation or Table Extension.
Rotations	Sets the number of tables to be maintained through the Duration. In this example no more than 8 tables will ever be maintained.

Due to overheads in the system, a setting of 7 rotations of 1 day yields between 4 and 5 days worth of data, depending on the time of day and time at which the rotation happens. To ensure 7 days worth of history, set the number of rotations to 10.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/platform-performance/concept/c_TableRotation.html

Table Extension

	Note: Deployment of this plugin should be executed in partnership with a ServiceNow representative.

Overview

Table Extension is a data archiving technique used to preserve instance performance and avert risk associated with querying growing data sets. To do this, Table Extension separates data sets into multiple tables based on date. The administrator of the process is given the ability to specify time parameter (Duration) of the process and number of tables (Rotations) within. However, altering the number of tables is strongly advised against. After the last table in a Rotation is written, new tables are added and old tables are archived. Using Table Extension, tables are never overwritten.

Advantages

- Partitions data across tables.
- Allows archiving of data while ensuring tables stay reasonably-sized.
- Reduces working set of data when date is known for the query.

Disadvantages

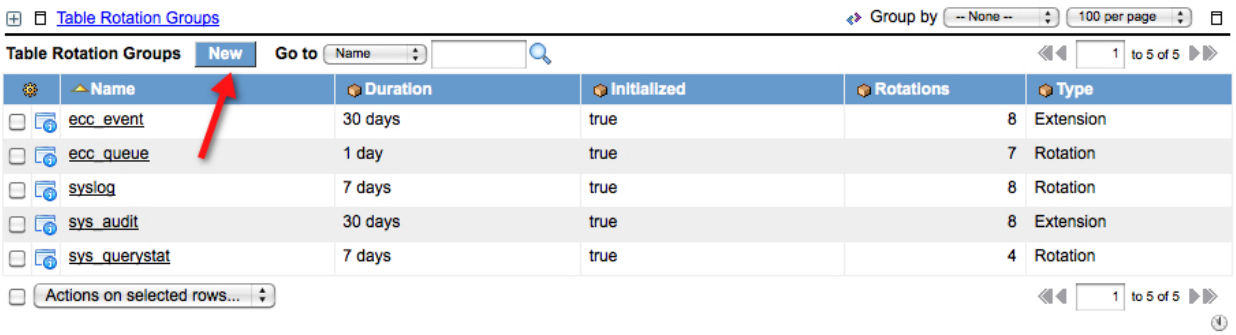
- Requires a union query when you query for a time range that spans multiple tables. Union queries are less efficient than queries against a single table.

Recommended Use

- Sequentially-written tables.
- Insert-only type tables (there are exceptions to this parameter).
- Tables where data is needed for long periods of time.

Example

To apply Table Extension, navigate to *System Definition* > *Table Rotation*. Choose *New*.

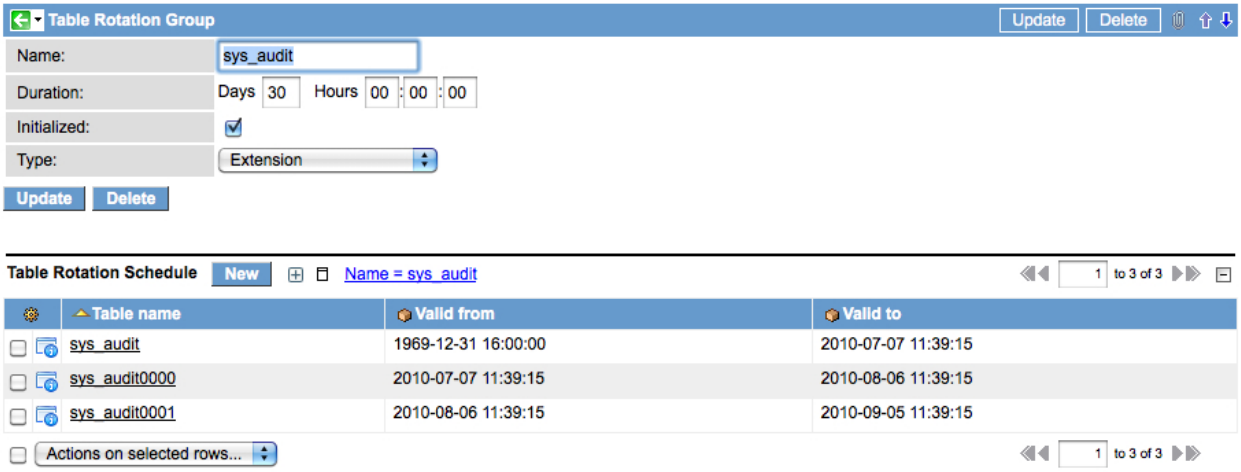


Once a new form is opened, set the following parameters:

1. **Name**: enter the name of the table to which the action is to be applied. In this case, the table is *sys_audit*.
2. **Duration**: set the overall duration for the action. 30 days is the duration in this example.
3. **Initialized**: sets function as active (true) or inactive (false).

4. **Type**: chooses either Table Rotation or Table Extension.

Selecting the *Update* button will initiate the action.



Exporting

Exporting Data

Note: *This article applies to Fuji and earlier releases. For more current information, see Exporting Data* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

Overview

ServiceNow offers a variety of ways for administrators and users with the itil role to export data:

- **Form export:** Export an individual record from the user interface. Choose PDF or XML format directly from a form.
- **List export:** Export multiple records from the user interface. Choose CSV, Excel, PDF, or XML format directly from a list.
- **Scheduled export with reports:** Automatically export multiple records from a table on a set schedule. Create a scheduled job to regularly export data as a report.
- **Direct URL access:** Export multiple records from a table using the ServiceNow CSV, Excel, PDF, or XML processor. Specify the table form or list you want to export in the URL.
- **Web services/SOAP:** Export multiple records from a table when an external client makes a web services request. Create an external application or process to automate the retrieval of data from an instance via web services or SOAP.

Available Export Formats

ServiceNow supports the following export formats:

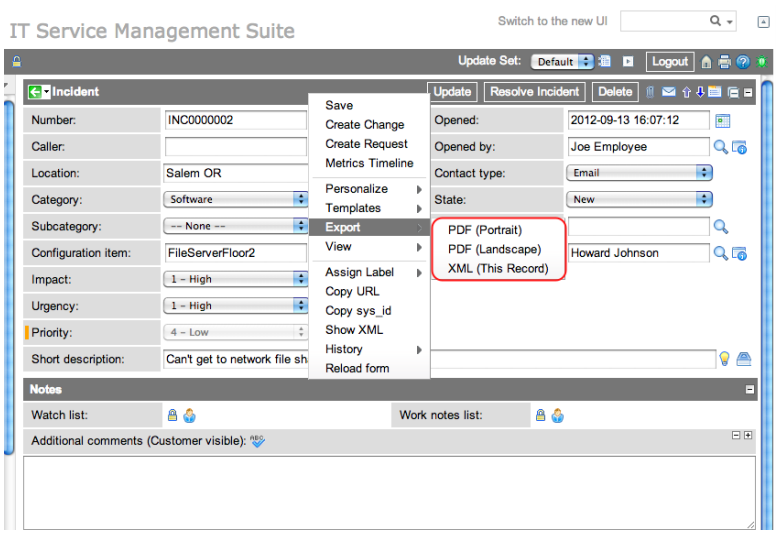
Export Format	Description
CSV	Export table records as a comma-separated value text file. Use this option to export the currently displayed fields in the list or form as a text file. Configure the list or form to add or remove fields from the export. When exported to CSV, dot-walked fields appear using their full field name, such as **u_assignment_group.parent**. **Note:** By default, ServiceNow exports all CSV files in Windows-1252 encoding. If you need to export translated data, set the `glide.export.csv.charset` system property to **UTF-8** (starting with Calgary release).
Excel	Export table records as a Microsoft Excel spreadsheet. Use this option to export the currently displayed fields in the list or form as an Excel spreadsheet. Configure the list or form to add or remove fields from the export.
XML	Export table records as an XML document. Use this option to export all data from a table or all data for a particular record. The XML file has an XML element for each column in the table.
PDF	Export table records as a Portable Document Format file. Use this option to export the currently displayed fields in the list or form as a PDF file. Configure the list or form to add or remove fields from the export.

For information about HTML and exported lists see Rendering HTML in Exported Lists [2] on the ServiceNow Community.

Exporting Form Data

Export an individual record from a form by right-clicking a form header bar and selecting the export type. Export formats include:

- PDF (Portrait)
- PDF (Landscape)
- XML (This Record)

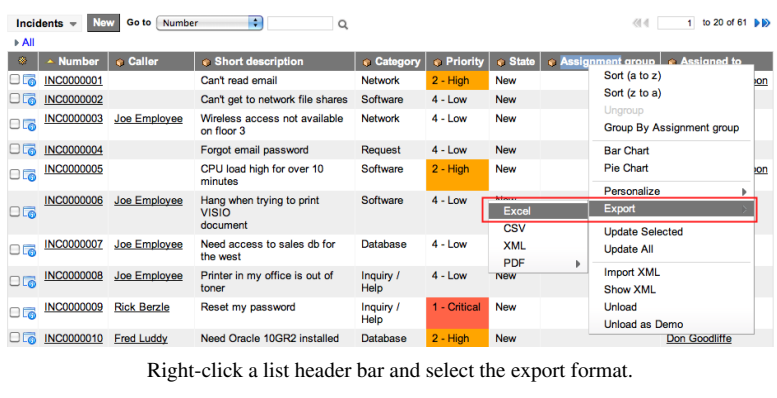


Right-click a form header bar and select the export format.

Note: *When exporting PDF data from a form, only the fields added to the current view are exported. Fields present on the form but hidden by a UI policy are included in the PDF. When exporting XML data, however, all the fields are exported, regardless of the view. CSV and Excel records cannot be exported from a form.*

Exporting List Data

Export a list of records by right-clicking a list header bar and selecting the export format.



Right-click a list header bar and select the export format.

Note: *To export records in an embedded list, export the record containing the list.*

Export formats include:

- Excel

- CSV
- XML
- PDF (Portrait)
- PDF (Landscape)
- PDF (Detailed Portrait): Exports the list and the associated form for each record.
- PDF (Detailed Landscape): Exports the list and the associated form for each record.

You can display Related List items on detailed PDF reports by setting the `glide.export.pdf.list.related_list` system property to `true`.

Note: *You can control how line breaks appear in exported CSV data using the `glide.csv.export.line_break` system property (starting with Dublin release).*

Determining Which List Fields are Exported

By default, exporting data from a list exports only the fields that are visible from the current view. If you want to export fields from another list view, you can switch views from the UI. Alternatively, you can add the `sysparm_view` parameter to the URL request. For example, to export fields visible from the Self Service (ess) view:

```
https://instance_name.service-now.com/incident.do?CSV&sysparm_view=ess
```

If you are exporting CSV or Excel data and do not specify a view, the export uses the default list view. You can export all fields by setting the `sysparm_default_export_fields` parameter to **all** (available starting with the Dublin release). If you are exporting XML data, the export returns all fields unless you specify a particular view. The `sysparm_default_export_fields` parameter has no effect on XML exports.

Exporting Data with Scheduled Reports

You can schedule regular or one-time exports from list reports. ServiceNow sends the report to one or more users by email. Export formats include:

- Excel
- CSV
- PDF
- PDF (Landscape)

If you need to share data with another ServiceNow instance or integrate with another application, consider creating a web service or SOAP message instead.

Exporting Directly from the URL

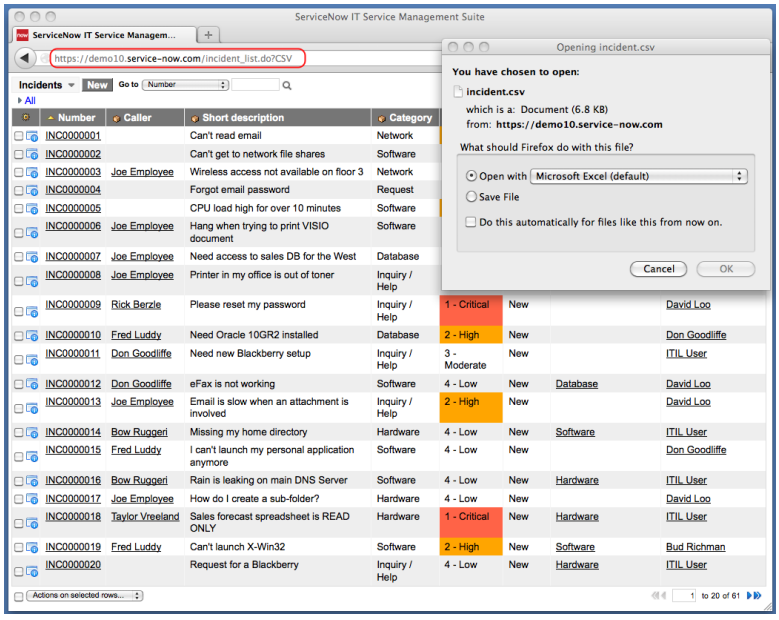
You may want to export data from the URL if you need to dynamically export data from a script or web service. You must be familiar with the ServiceNow table and column names to export data directly from the URL. See Navigating by URL for more information about navigating to forms and lists.

To export data directly from the URL, create a URL containing the following parts:

1. Specify the instance URL. For example, **https://<instance name>.service-now.com/**.
2. Specify the table form or list to export. For example, **incident_list.do**.
3. Specify the export format processor to use for the export. For example, **?CSV**.
4. [Optional] Specify a query and sort order with URL parameters. For example, **&sysparm_query=sys_id%3E%3Db4aedb520a0a0b1001af10e278657d27**.

The final URL should look like one of these sample URLs:

URL	Description
https://<instance name>.service-now.com/incident_list.do?CSV	Export all incidents to a comma-separated value text file.
https://<instance name>.service-now.com/incident_list.do?CSV&sysparm_query=sys_id%3E%3Db4aedb520a0a0b1001af10e278657d27	Export a particular incident to a comma-separated value text file.
https://<instance name>.service-now.com/incident_list.do?CSV&sysparm_order_by=sys_id	Export all incidents to a comma-separated value text file and sort the list by sys_id.



Exporting data directly from the URL

Note: *ServiceNow enforces basic authentication* [3] *for direct URL access. The data extracted from the URL contains only the fields to which the currently authenticated user has read access.*

Export Format Processors

The ServiceNow platform provides a default upper limit for XML data exports starting with the Dublin release. If you are using an older version, see the previous version information.

ServiceNow provides the following export format processors:

Export Processor	URL Syntax	Export Limits	Description
CSV	?CSV	10,000 rows	Exports table records as a comma-separated value text file.
Excel	?EXCEL	10,000 rows	Exports table records as a Microsoft Excel spreadsheet.
XML	?XML	10,000 rows	Exports table records as an XML document.
PDF	?PDF	5,000 rows	Exports table records as a Portable Document Format file.
Schema	?SCHEMA	N/A	Exports the database schema for the table.
XSD	?XSD	N/A	Exports the table structure in XSD format.

See Export Limits for information about processor export limits and how to work around them if a table exceeds the export limit.

Note: *Export processors return all requested records regardless of whether you use the* ***table.do*** *or* ***table_list.do*** *format to identify the export table.*

Click the plus to expand previous version information

Prior to the Dublin release, the ServiceNow platform provided default upper and lower limits for XML data exports.

Note: *If you use these limits in a Dublin instance, they will continue to function correctly. To ensure functionality in subsequent releases, however, we recommend that you modify your code to use the single export limit.*

Export Processor	URL Syntax	Export Limits	Description
CSV	?CSV	**Lower Limit**: 10,000 rows **Upper Limit**: 50,000 rows	Export table records as a comma-separated value text file.
Excel	?EXCEL	**Lower Limit**: 10,000 rows **Upper Limit**: 50,000 rows	Export table records as a Microsoft Excel spreadsheet.
XML	?XML	**Lower Limit**: 10,000 rows **Upper Limit**: 50,000 rows	Export table records as an XML document.
PDF	?PDF	**Upper Limit**: 5,000 rows	Export table records as a Portable Document Format file.
Schema	?SCHEMA	**Lower Limit**: 10,000 rows **Upper Limit**: 50,000 rows	Export table records as a schema document.
XSD	?XSD	**Lower Limit**: 10,000 rows **Upper Limit**: 50,000 rows	Export table records as an XSD document.

Using URL Queries to Filter List Results

Use URL queries to programmatically generate filtered lists before exporting them. URL queries are useful for scripts that generate a list of records and where no user will manually add the filter from the UI. You must be familiar with the table's column names and values to create a query.

To create URL queries:

1. Specify the instance URL. For example, **https://demo.service-now.com/**.
2. Specify the list URL for the table you want to export. For example, **incident_list.do**.
3. Specify the export format processor for the export. For example, **?XML**.
4. Specify the query as the value of the `sysparm_query` parameter. For example, **?sysparm_query=priority=1**.
5. [Optional] Specify the result sort order with the `sysparm_order_by=` parameter. For example, **CSV&sysparm_order_by=assigned_to**.

Note: *All queries use a column name, an operator, and a value. See Condition Builder for a list of available operators.*

For example, to export a list of all priority 1 incidents as an XML file, use the following URL:
https://<instance name>.service-now.com/incident_list.do?XML&sysparm_query=priority=1&sysparm_order_by=assigned_to

URL Query Parameters

ServiceNow provides the following URL query parameters:

URL Parameter	URL Syntax	Description
sysparm_query	sysparm_query=[column name][operator][value]	Displays a list of records that match the query. For example: **https://<instance name>.service-now.com/incident_list.do?XML&sysparm_query=priority=1**
sysparm_orderby	sysparm_orderby=[column name]	Sorts a list of records by the column name provided. For example: **https://<instance name>.service-now.com/incident_list.do?sysparm_query=priority=1&sysparm_orderby=assigned_to** You can sort by only one column using sysparm_orderby. To sort by multiple columns, use sysparm_query=ORDERBY[column name]^ORDERBY[column name]. For example: **sysparm_query=ORDERBYassigned_to^ORDERBYpriority**.

Example XML Export

The following URL query produces an XML document similar to the sample shown:

https://<instance name>.service-now.com/incident_list.do?XML&sysparm_query=priority=1&sysparm_order_by=assigned_to

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
<incident>
<active>true</active>
<activity_due>2012-11-28 20:44:11</activity_due>
<approval>not requested</approval>
<approval_history/>
<approval_set/>
```

```
<assigned_to/>
<assignment_group>d625dccec0a8016700a222a0f7900d06</assignment_group>
<business_duration/>
<business_stc/>
<calendar_duration/>
<calendar_stc/>
<caller_id/>
<category>inquiry</category>
<caused_by/>
<child_incidents>0</child_incidents>
<close_code/>
<close_notes/>
<closed_at/>
<closed_by/>
<cmdb_ci/>
<comments/>
<comments_and_work_notes/>
<company/>
<contact_type>phone</contact_type>
<correlation_display/>
<correlation_id/>
<delivery_plan/>
<delivery_task/>
<description/>
<due_date/>
<escalation>0</escalation>
<expected_start/>
<follow_up/>
<group_list/>
<impact>1</impact>
<incident_state>1</incident_state>
<knowledge>false</knowledge>
<location>108752c8c611227501d4ab0e392ba97f</location>
<made_sla>true</made_sla>
<notify>1</notify>
<number>INC0010040</number>
<opened_at>2012-11-28 16:31:08</opened_at>
<opened_by>6816f79cc0a8016401c5a33be04be441</opened_by>
<order/>
<parent/>
<parent_incident/>
<priority>1</priority>
<problem_id/>
<reassignment_count>0</reassignment_count>
<rejection_goto/>
<reopen_count>0</reopen_count>
<resolved_at/>
```

```
<resolved_by/>
<rfc/>
<severity>3</severity>
<short_description>test +XB</short_description>
<skills/>
<sla_due/>
<state>1</state>
<subcategory/>
<sys_class_name>incident</sys_class_name>
<sys_created_by>admin</sys_created_by>
<sys_created_on>2012-11-28 16:31:33</sys_created_on>
<sys_domain>global</sys_domain>
<sys_id>4efe74640982300000a286aa871e705a</sys_id>
<sys_mod_count>4</sys_mod_count>
<sys_updated_by>admin</sys_updated_by>
<sys_updated_on>2012-11-28 18:44:11</sys_updated_on>
<time_worked/>
<upon_approval>proceed</upon_approval>
<upon_reject>cancel</upon_reject>
<urgency>1</urgency>
<user_input/>
<variables/>
<watch_list>dfdc91a809c2300000a286aa871e7057,46d44a23a9fe19810012d100cca80666</watch_list>
<wf_activity/>
<work_end/>
<work_notes/>
<work_notes_list/>
<work_start/>
</incident>
...
</xml>
```

Calling URL Exports Programmatically

You can dynamically export data from a script or web service by calling a URL export from any programming language. The following procedure includes code samples that demonstrate calling a URL export in C# for a .Net framework call:

1. Add the following Imports:

```
using System.Net;
using System.IO;
```

2. Call the Download method:

```
        static void Main(string[] args)


    {
        // Call to DownloadFile method supplying the URL and location
to save CSV file locally
```

```
        int read =
DownloadFile("https://demo007.service-now.com/incident_list.do?CSV&sysparm_query=priority=1&sysparm_order_by=assigned_to",
                    "c:\\test\\incident.csv");
    }
```

3. Create a Download method as follows:

```
        public static int DownloadFile(String url,
                                        String localFilename)
    {
        // Function will return the number of bytes processed
        // to the caller. Initialize to 0 here.
        int bytesProcessed = 0;
        // Assign values to these objects here so that they can
        // be referenced in the finally block
        Stream remoteStream = null;
        Stream localStream = null;
        WebResponse response = null;
        // Use a try/catch/finally block as both the WebRequest and
Stream
        // classes throw exceptions upon error
        try
        {
            // Create a request for the specified remote file name
            WebRequest request = WebRequest.Create(url);
            // Create the credentials required for Basic Authentication
            System.Net.ICredentials cred = new
System.Net.NetworkCredential("user_name", "password");
            // Add the credentials to the request
            request.Credentials = cred;
            if (request != null)
            {
                // Send the request to the server and retrieve the
                // WebResponse object
                response = request.GetResponse();
                if (response != null)
                {
                    // Once the WebResponse object has been retrieved,
                    // get the stream object associated with the
response's data
                    remoteStream = response.GetResponseStream();
                    // Create the local file
                    localStream = File.Create(localFilename);
                    // Allocate a 1k buffer
                    byte[] buffer = new byte[1024];
                    int bytesRead;
                    // Simple do/while loop to read from stream until
                    // no bytes are returned
```

```
                    do
                    {
                        // Read data (up to 1k) from the stream
                        bytesRead = remoteStream.Read(buffer, 0,
buffer.Length);
                        // Write the data to the local file
                        localStream.Write(buffer, 0, bytesRead);
                        // Increment total bytes processed
                        bytesProcessed += bytesRead;
                    } while (bytesRead > 0);
                }
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
        finally
        {
            // Close the response and streams objects here
            // to make sure they're closed even if an exception
            // is thrown at some point
            if (response != null) response.Close();
            if (remoteStream != null) remoteStream.Close();
            if (localStream != null) localStream.Close();
        }
        // Return total bytes processed to caller.
        return bytesProcessed;
    }
```

Export Limits

The ServiceNow platform provides a default upper limit for XML data exports.

The purpose of the upper limit is to avoid creating performance issues when a table is excessively large. If you need to export more records than the threshold permits, break up the export into separate manageable chunks.

In addition to the format-specific limits, you may need to set `com.glide.processors.XMLProcessor.max_record_count` to match the upper limit set by the format-specific limit.

Export Limit Properties

You can set the number of records to return during an export using the URL parameter `sysparm_record_count`. However, the system analyzes the following settings to determine whether an export limit should be applied.

1. First, the platform checks the property that defines the format-specific export limit (see table). Each format can have a different limit. Although this property can be set to any value, exceeding the default export limit can impact system performance. You may want to set the property at or below the default limit and have users export large amounts of data in smaller increments.
2. If the format-specific property is not set, the system checks the property for the general export limit (see table). This property can also be set to any value, but exceeding the default export limit can impact system performance.
3. If neither the format-specific export limit nor the general export limit property is set, the system enforces the default export limit (see table).

Note: *These properties are not defined by default. You must add the property to assign a value to it.*

Format	Format-Specific Export Limit	General Export Limit	Default Export Limit
XML	glide.xml.export.limit	glide.ui.export.limit	10,000
CSV	glide.csv.export.limit	glide.ui.export.limit	10,000
EXCEL	glide.excel.export.limit	glide.ui.export.limit	10,000
PDF	glide.pdf.max_rows	N/A	5,000
PDF	glide.pdf.max_columns	N/A	25

Although the number of columns can be set higher than 25 in the PDF export, this is not advisable as only 25 header labels fit on a page.

A warning threshold property called `glide.ui.export.warn.threshold` controls how the records are exported. If a user attempts to export a number of records from a list that exceeds the warning threshold, a dialog box offers the choice of waiting for the export to complete or having the exported records emailed as an attachment. The warning threshold can be changed in the system property. The email attachment must not exceed the maximum allowed email size or configured email attachment size. For more information, see Exported Table Emails.

Example 1: Exporting to CSV

- `glide.csv.export.limit` = 20,000
- `glide.ui.export.limit` = 10,000
- `com.glide.processors.XMLProcessor.max_record_count` = 20,000
- Default export limit for CSV = 10,000

Records to be Exported	Records Returned
15,000	15,000
30,000	20,000

Note: *In the second export, the number of records returned from the database is limited because the number of records specified for export exceeds the value set in the* `glide.csv.export.limit` *property.*

Example 2: Exporting to Excel

- `glide.excel.export.limit` = no entry
- `glide.ui.export.limit` = no entry
- Default export limit for EXCEL = 10,000

Records to be Exported	Records Returned
10,000	10,000
30,000	10,000

Note: *In the second export, the number of records returned from the database is limited because the number of records specified for export exceeds the default export limit for Excel, 10,000 records.*

Example 3: Exporting to PDF

- `glide.pdf.max_rows` = 3,500
- Default and maximum export limit for PDF = 5,000

Records to be Exported	Records Returned
2,000	2,000
10,000	5,000

Note: *In the first export, all records are returned because the number of records specified for export does not exceed the* `glide.pdf.max_rows` *property. In the second export, the number of records returned is limited because the number of records specified for export exceeds the value in the* `glide.pdf.max_rows` *property.*

Breaking Up Large Exports

If the number of records to be exported exceeds the actual export limit, you may want to break the export into smaller increments that do not place a significant performance load on the platform.

1. Filter the list to display the records you want to export.
2. Write down the number of records returned.
3. If the record number is higher than the defined threshold, issue a `sysparm` query for the first 10,000 records using the following syntax:

   https://<instance name>.service-now.com/syslog_list.do?XML&sysparm_order_by=sys_id&sysparm_record_count=10000

   This exports the first 10,000 records in order, sorted by the **sys_id** number.

4. Find the next record in order, such as 10,001.
5. Right-click the row and copy the sys_id of the next record you want to export.
6. Access the next series of records with a **greater than or equal to** query run against the **sys_id** of record 10,001.

   The following example shows a query that uses a sys_id of *b4aedb520a0a0b1001af10e278657d27*. Use the syntax shown in this query to export the next set of records.

   https://<instance name>.service-now.com/syslog_list.do?XML&sysparm_query=sys_id%3E%3D*b4aedb520a0a0b1001af10e278657d27***&**

   Note: URL queries use typical percent encoding [4]. In this example, the greater than sign (>) is encoded as %3E and the equal sign (=) is encoded as %3D.
7. Continue issuing this query, using the starting sys_id for the next set of records until you have exported all the necessary records.

Excel Export Threshold

Excel exports are intended for relatively small exports, fewer than 500,000 cells, while CSV can handle larger exports.

Whenever you export to Excel and the resultant spreadsheet has more than 500,000 cells (by default), the export process stops and you are given the Excel file at that point. In the bottom row, there will be the following message: **Export stopped due to excessive size. Use CSV for a complete export**:

	A	B	C	D	
1	**Table**	**Column name**	**Text index**	**Calculated**	**Re**
16848	cmdb_tcp_half	sys_mod_count	FALSE	FALSE	
16849	cmdb_tcp_half	sys_created_on	FALSE	FALSE	
16850	cmdb_tcp_half	sys_created_by	FALSE	FALSE	
16851	**Export stopped due to excessive size. Use CSV for a complete export.**				
16852					
16853					

The Excel export cell threshold is customizable using the glide.excel.max_cells integer property. **Note:** Increasing this threshold may cause a memory issue in your instance. The threshold is set at an appropriate level to prevent resource issues.

The export will put the information into the Excel document with 32,000 rows per spreadsheet.

Enabling Export Debug Logging

When the property `glide.export.debug` is **true**, the instance logs export processing including database query time and the time taken to write data to the file. Debug logs are indicated by the text **Export API**. Prolonged use of this property can affect performance, so it is best to use it while debugging export processing, and then set the property back to **false**.

Click the plus to see sample log data.

```
7/17/14 15:53:48 (500) EB39A310EB022100C46AC2EEF106FED9 Maximum record
count for this instance is: 10000: , request is for: 0 Cap the Record
count to Maximum Record Count
07/17/14 15:53:48 (522) EB39A310EB022100C46AC2EEF106FED9 Export API -
ExportProcessor : Processing EXCEL export request,
ExportParameters:TableName=incident, Query=active=true, Limit=10000,
SortBy=null
07/17/14 15:53:48 (527) EB39A310EB022100C46AC2EEF106FED9 Export API -
ExportProcessor : Export using background thread
07/17/14 15:53:48 (528) EB39A310EB022100C46AC2EEF106FED9 #748
/poll_processor.do -- total transaction time: 0:00:03.357, total wait
time: 0:00:00.000, session wait: 0:00:00.000, semaphore wait:
0:00:00.000, source: 0:0:0:0:0:0:0:1%0
07/17/14 15:53:48 (550) SYSTEM Enabling elevated role: security_admin
07/17/14 15:53:48 (588) SYSTEM Export API - ExcelExporter : 29 rows
retrieved from database duration_milliseconds=2
07/17/14 15:53:49 (534) NONE New transaction
EB39A310EB022100C46AC2EEF106FED9 #751 /poll_processor.do
07/17/14 15:53:49 (544) EB39A310EB022100C46AC2EEF106FED9 #751
/poll_processor.do Parameters -------------------------
    sys_action=poll
    sysparm_processor=poll_processor
    job_id=a61a2314eb022100c46ac2eef106fe0a
07/17/14 15:53:49 (547) EB39A310EB022100C46AC2EEF106FED9 #751
/poll_processor.do -- total transaction time: 0:00:00.013, total wait
time: 0:00:00.000, session wait: 0:00:00.000, semaphore wait:
0:00:00.000, source: 0:0:0:0:0:0:0:1%0
07/17/14 15:53:49 (740) SYSTEM Export API - ExcelExporter : Rows
written to file duration_milliseconds=1150
```

Enhancements

Fuji

You can enable export logging to record detailed export performance data.

Dublin

- A new system property, `glide.csv.export.line_break`, controls how line breaks appear in exported CSV data.
- A default upper limit is available for XML data exports. However, the upper and lower limits from earlier versions will continue to function correctly in a Dublin instance.
- A new system property, `glide.ui.export.warn.threshold`, controls how a user receives returned export records when the limit has been exceeded. The available options are to wait for the export to complete or have the exported records sent as an email attachment.

Calgary

- A new system property, `glide.export.csv.charset`, controls the character set used to export CSV files. This property allows localized instances to export strings in a character set supported by the language. Starting with Calgary, you can set the system property to the character set name you want to use to export CSV files.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/exporting-data/concept/c_ExportData.html
[2] https://community.servicenow.com/community/blogs/blog/2015/01/05/rendering-html-in-exported-lists
[3] http://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html#BasicAA
[4] http://en.wikipedia.org/wiki/Percent-encoding

Article Sources and Contributors

Introduction to Managing Data *Source*: http://wiki.servicenow.com/index.php?oldid=251085 *Contributors*: Cheryl.dolan, David.Bailey, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.ramos, Joseph.messerschmidt, Neola, Rachel.sienko, Roy.lagemann, Vaughn.romero

Data Dictionary Tables *Source*: http://wiki.servicenow.com/index.php?oldid=250479 *Contributors*: Emily.partridge, Fuji.publishing.user, G.yedwab, John.ramos, Joseph.messerschmidt, Neola, Rachel.sienko

System Dictionary *Source*: http://wiki.servicenow.com/index.php?oldid=250373 *Contributors*: Amy.bowman, Cheryl.dolan, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, Jim.uebbing, John.ramos, Joseph.messerschmidt, Neola, Phillip.salzman, Rachel.sienko, Steven.wood, Suzanne.smith, Vaughn.romero

Dictionary Attributes *Source*: http://wiki.servicenow.com/index.php?oldid=250505 *Contributors*: Amy.bowman, CapaJC, Cheryl.dolan, Chuck.tomasi, Emily.partridge, Eric.jacobson, Fuji.publishing.user, G.yedwab, Guy.yedwab, Jim.uebbing, John.ramos, John.roberts, Joseph.messerschmidt, Michael.randall, Neil.narvaez, Neola, Phillip.salzman, Rachel.sienko, Steven.wood, Tom.dilatush, Vaughn.romero

Dictionary Overrides *Source*: http://wiki.servicenow.com/index.php?oldid=250506 *Contributors*: Cheryl.dolan, Chuck.tomasi, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, Joe.zucker, John.ramos, Joseph.messerschmidt, Katharine.sohler, Neola, Rachel.sienko, Steven.wood, Vaughn.romero, Wallymarx

Archiving Data *Source*: http://wiki.servicenow.com/index.php?oldid=250127 *Contributors*: Cheryl.dolan, John.ramos, John.roberts, Phillip.salzman, Rachel.sienko, Vaughn.romero

Database Rotation *Source*: http://wiki.servicenow.com/index.php?oldid=207459 *Contributors*: G.yedwab, Gadi.yedwab, Guy.yedwab, Joseph.messerschmidt, Neola, Rachel.sienko, Vaughn.romero, Wallymarx

Table Rotation *Source*: http://wiki.servicenow.com/index.php?oldid=250938 *Contributors*: Anat.kerry, Emily.partridge, G.yedwab, Gadi.yedwab, John.ramos, Joseph.messerschmidt, Neola, Rachel.sienko, Vaughn.romero, Wallymarx

Table Extension *Source*: http://wiki.servicenow.com/index.php?oldid=101523 *Contributors*: G.yedwab, Gadi.yedwab, Joseph.messerschmidt, Neola, Rachel.sienko, Wallymarx

Exporting Data *Source*: http://wiki.servicenow.com/index.php?oldid=250574 *Contributors*: Aburruss, Amy.bowman, CapaJC, Christen.mitchell, David Loo, David.Bailey, Dkearney, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.ramos, John.roberts, Joseph.messerschmidt, Ludwig.adriaansen, Mark.stanger, Neola, Roy.lagemann, Steven.wood, Suzanne.smith, Vaughn.romero, Vhearne

Image Sources, Licenses and Contributors

Image:Warning.gif *Source*: http://wiki.servicenow.com/index.php?title=File:Warning.gif *License*: unknown *Contributors*: CapaJC

Image:DictionaryIncTask.png *Source*: http://wiki.servicenow.com/index.php?title=File:DictionaryIncTask.png *License*: unknown *Contributors*: G.yedwab

Image:Fieldlabels.png *Source*: http://wiki.servicenow.com/index.php?title=File:Fieldlabels.png *License*: unknown *Contributors*: G.yedwab

Image:Tables.png *Source*: http://wiki.servicenow.com/index.php?title=File:Tables.png *License*: unknown *Contributors*: Rachel.sienko

Image:dictionary_entry_form_Eureka.png *Source*: http://wiki.servicenow.com/index.php?title=File:Dictionary_entry_form_Eureka.png *License*: unknown *Contributors*: Maintenance script, Phillip.salzman

Image:archiving_benefits.png *Source*: http://wiki.servicenow.com/index.php?title=File:Archiving_benefits.png *License*: unknown *Contributors*: Vaughn.romero

Image:system_archiving_app.png *Source*: http://wiki.servicenow.com/index.php?title=File:System_archiving_app.png *License*: unknown *Contributors*: Vaughn.romero

Image:create_archive_rule_01.png *Source*: http://wiki.servicenow.com/index.php?title=File:Create_archive_rule_01.png *License*: unknown *Contributors*: Phillip.salzman, Vaughn.romero

Image:create_archive_rule_02.png *Source*: http://wiki.servicenow.com/index.php?title=File:Create_archive_rule_02.png *License*: unknown *Contributors*: Vaughn.romero

Image:flatten_tables.png *Source*: http://wiki.servicenow.com/index.php?title=File:Flatten_tables.png *License*: unknown *Contributors*: Vaughn.romero

Image:localized_archives.png *Source*: http://wiki.servicenow.com/index.php?title=File:Localized_archives.png *License*: unknown *Contributors*: Vaughn.romero

Image:module_for_archived_table.png *Source*: http://wiki.servicenow.com/index.php?title=File:Module_for_archived_table.png *License*: unknown *Contributors*: Vaughn.romero

Image:list_archived_table.png *Source*: http://wiki.servicenow.com/index.php?title=File:List_archived_table.png *License*: unknown *Contributors*: Vaughn.romero

Image:form_archived_table.png *Source*: http://wiki.servicenow.com/index.php?title=File:Form_archived_table.png *License*: unknown *Contributors*: Vaughn.romero

Image:archive_log.png *Source*: http://wiki.servicenow.com/index.php?title=File:Archive_log.png *License*: unknown *Contributors*: Vaughn.romero

Image:records_restored.png *Source*: http://wiki.servicenow.com/index.php?title=File:Records_restored.png *License*: unknown *Contributors*: Vaughn.romero

Image:Caution-diamond.png *Source*: http://wiki.servicenow.com/index.php?title=File:Caution-diamond.png *License*: unknown *Contributors*: John.roberts, Publishing.user

Image:archive_restored_record.png *Source*: http://wiki.servicenow.com/index.php?title=File:Archive_restored_record.png *License*: unknown *Contributors*: Vaughn.romero

File:Database Rotation.jpg *Source*: http://wiki.servicenow.com/index.php?title=File:Database_Rotation.jpg *License*: unknown *Contributors*: Wallymarx

File:Database Rotation new arrow.jpg *Source*: http://wiki.servicenow.com/index.php?title=File:Database_Rotation_new_arrow.jpg *License*: unknown *Contributors*: Wallymarx

File:Table Rotation .jpg *Source*: http://wiki.servicenow.com/index.php?title=File:Table_Rotation_.jpg *License*: unknown *Contributors*: Wallymarx

File:Table Extension .jpg *Source*: http://wiki.servicenow.com/index.php?title=File:Table_Extension_.jpg *License*: unknown *Contributors*: Wallymarx

Image:export_from_form.png *Source*: http://wiki.servicenow.com/index.php?title=File:Export_from_form.png *License*: unknown *Contributors*: Vaughn.romero

Image:export_from_list.png *Source*: http://wiki.servicenow.com/index.php?title=File:Export_from_list.png *License*: unknown *Contributors*: Vaughn.romero

Image:export_from_url.png *Source*: http://wiki.servicenow.com/index.php?title=File:Export_from_url.png *License*: unknown *Contributors*: Vaughn.romero

Image:ExcelExportThreshold1.png *Source*: http://wiki.servicenow.com/index.php?title=File:ExcelExportThreshold1.png *License*: unknown *Contributors*: CapaJC