

# Complete Documentation and Configuration Guide

**Project Title:** Automated Car Catalog System for Enhanced Showroom Management

Team ID : LTVIP2025TMID31086

Team Size : 4

Team Leader : Gubbala Subrahmanya Gopichand

Team member : Guddati Sriram

Team member : Kalvakolanu Lakshmi Swathi

Team member : Kumpatla Devi Priya Darsana

**College Name:** *Bonam Venkata chalamayya institute of technology and science*

**Project Duration:** *8 hours*

**Internship Organization:** *SmartBridge*

---

## Executive Summary

This document provides comprehensive documentation for the ServiceNow Family Expenses Management System, including table structures, business rules, relationships, and form configurations. The system is designed to track and manage family expenditures with automated number generation and expense aggregation capabilities.

---

## Table of Contents

1. [System Overview](#)
  2. [Table Structures](#)
  3. [Form Configurations](#)
  4. [Business Rules](#)
  5. [Relationships](#)
  6. [Number Management](#)
  7. [Related Lists Configuration](#)
  8. [Technical Implementation](#)
- 

## System Overview

The Family Expenses Management System consists of two primary tables:

- **Family expenses** (u\_family\_expenses) - Master table for family member information

- **Daily expenses** (u\_daily\_expenses) - Detail table for individual expense records

The system implements a parent-child relationship where family members can have multiple daily expense entries, with automated expense aggregation and number generation.

The screenshot shows the ServiceNow 'Relationship - New Record' form. The 'Name' field is set to 'Daily Expenses'. The 'Application' is 'Global'. The 'Applies to table' is 'Family Expenses [u\_st\_family\_expenses]'. The 'Queries from table' is '-- None --'. Below these fields is a script editor with a pre-defined function `refineQuery(current, parent)` and a comment to add code. A 'Submit' button is at the bottom.

## Table Structures

### 1. Family Expenses Table (u\_family\_expenses)

**Purpose:** Master table containing family member information and aggregated expense data.

#### Key Fields:

- **Number** (String) - Auto-generated identifier using MFE prefix
- **Date** (Date) - Record creation date
- **Amount** (Integer) - Aggregated total of all related daily expenses
- **Expenses Details** (String) - Concatenated details of all related expenses
- **Family member name** (Reference) - Links to family member records
- **Comments** (String) - Additional notes and comments

#### System Fields:

- Sys ID (GUID)
- Created (Date/Time)
- Updated by (String)
- Updates (Integer)
- Updated (Date/Time)
- Created by (String)

Family expenses [u\_family\_expenses] | Default view | Form Design

Fields | Field Types

Filter

Fields

- Created
- Created by
- Updated
- Updated by
- Updates

Formatters

- Activities (filtered)
- Contextual Search Results
- Ratings

Family expenses [u\_family\_expenses] | 2 Column

- Number
- Amount
- Date

Expenses Details | 1 Column

## 2. Daily Expenses Table (u\_daily\_expenses)

**Purpose:** Detail table for individual expense entries linked to family members.

### Key Fields:

- **Number** (String) - Auto-generated expense record identifier
- **Date** (Date) - Expense transaction date
- **Expenses** (Integer) - Individual expense amount
- **Comments** (String) - Expense description and notes
- **Family member name** (Reference) - Links to u\_family\_expenses table

### System Fields:

- Standard ServiceNow audit fields (Created, Updated, etc.)

servicenow All Favorites History Workspaces Admin Table - Daily expenses Search

Table Daily expenses Delete Update Delete All Records

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)

\* Label Daily expenses Application Global

\* Name u\_daily\_expenses

Columns Controls Application Access

Table Columns for text Search 1 to 6 of 6 New

Column label	Type	Reference	Max length	Default value	Display
Sys ID	Sys ID (GUID)	(empty)	32		false
Created	Date/Time	(empty)	40		false
Updated by	String	(empty)	40		false
Updates	Integer	(empty)	40		false
Updated	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false

+ Insert a new row...

## Form Configurations

### Family Expenses Form Layout

The form is configured with a **2-column layout** containing:

#### Top Section (2 columns):

- Number field (left column)
- Amount field (right column)
- Date field (spans across)
- Expenses Details field (full width)

#### Bottom Section (1 column):

- Comments field (full width)

### Daily Expenses Form Layout

The form uses a **2-column layout** with:

#### Top Section (2 columns):

- Number field (left column)
- Date field (right column)
- Expenses field (spans across)

#### Bottom Section (1 column):

- Comments field (full width)

## Business Rules

### Family Expenses Business Rule (Family expenses BR)

#### Configuration:

- **Name:** Family expenses BR
- **Table:** Daily expenses [u\_daily\_expenses]
- **Active:** Yes
- **Advanced:** Yes
- **ECMAScript 2021 (ES12) mode:** Enabled

#### Script Logic:

javascript

```
(function executeRule(current, previous /*null when async*/) {
```

```

// Initialize GlideRecord for Family expenses table

var FamilyExpenses = new GlideRecord('u_family_expenses');


// Query for matching date records

FamilyExpenses.addQuery('u_date', current.u_date);

FamilyExpenses.query();


// If matching record exists, update it

if (FamilyExpenses.next()) {

    // Aggregate expense amount

    FamilyExpenses.u_amount += current.u_expense;


    // Concatenate expense details

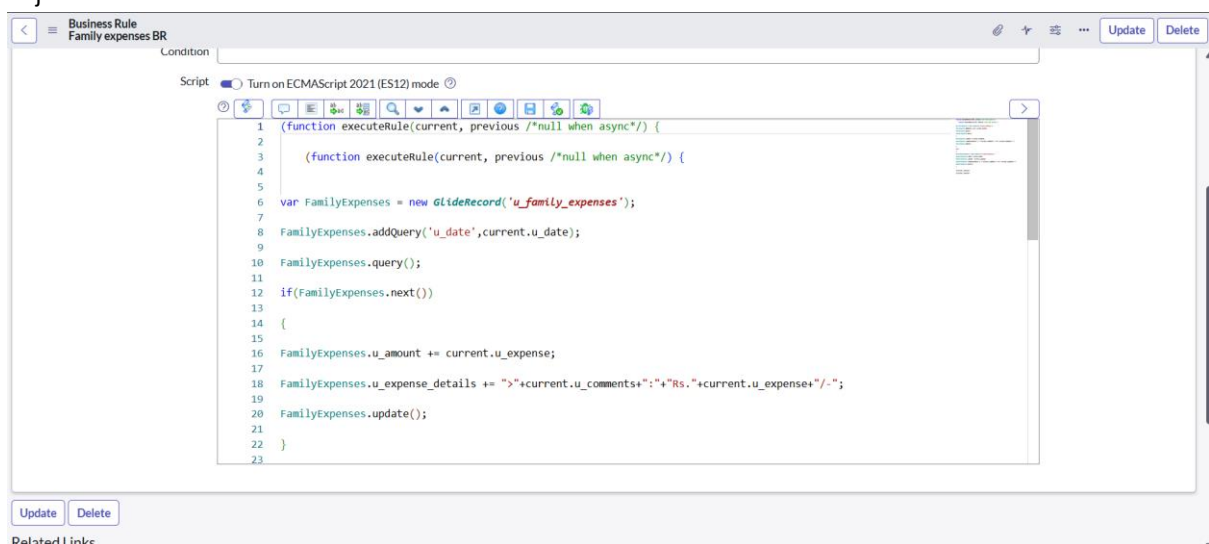
    FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." +
current.u_expense + "/-";


    // Update the record

    FamilyExpenses.update();

}

```



```

})(current, previous);

```

### Functionality:

1. Automatically aggregates daily expenses into family expense totals

2. Concatenates expense details with formatting
3. Updates parent records when child records are created/modified
4. Maintains data consistency between related tables

## Relationships

### Daily Expenses Relationship

#### Configuration:

- **Name:** Daily expenses
- **Applies to table:** Family expenses [u\_family\_expenses]
- **Queries from table:** Daily expenses [u\_daily\_expenses]
- **Advanced:** No

servicenow All Favorites History Workspaces Relationship - New Record

Name: Daily Expenses Application: Global

Advanced: ☐ Applies to table: Family Expenses [u\_st\_family\_expenses]

Queries from table: -- None --

This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see the documentation. See also the article about the recommended form of the script.

Query with ☐ Turn on ECMAScript 2021 (ES12) mode

```
1 (function refineQuery(current, parent) {
2
3 // Add your code here, such as current.addQuery(field, value);
4
5 })(current, parent);
```

Submit

servicenow All Favorites History Workspaces Relationship - New Record

Name: Daily Expenses Application: Global

Advanced: ☐ Applies to table: Family Expenses [u\_st\_family\_expenses]

Queries from table: -- None --

This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see the documentation. See also the article about the recommended form of the script.

Query with ☐ Turn on ECMAScript 2021 (ES12) mode

```
1 (function refineQuery(current, parent) {
2
3 // Add your code here, such as current.addQuery(field, value);
4
5 })(current, parent);
```

Submit

## Query Script:

javascript

```
(function refineQuery(current, parent) {  
    // Add date-based query condition  
    current.addQuery('u_date', parent.u_date);  
    current.query();  
})(current, parent);
```

**Purpose:** Links daily expense records to their corresponding family expense parent records based on date matching.

Relationship  
Daily expenses

Name

Advanced ☐

Application

Applies to table

Queries from table

This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see [the documentation](#). See also the article about the [recommended form of the script](#).

Query with ☒ Turn on ECMAScript 2021 (ES12) mode

```
1 (function refineQuery(current, parent) {  
2  
3  
4 // Add your code here, such as current.addQuery(field, value);  
5  
6 current.addQuery('u_date', parent.u_date);  
7  
8 current.query();  
9  
10  
11 })(current, parent);  
12
```

Related Links  
[Run Point Scan](#)

## Number Management

### Number Generation Configuration (Number - MFE)

#### Settings:

- **Table:** Family expenses
- **Prefix:** MFE
- **Number:** 1,000 (starting number)
- **Application:** Global
- **Number of digits:** 7

**Format:** MFExxxxxxx (where x represents sequential digits)

#### Examples:

- MFE0001000
- MFE0001001



- MFE0001002

The screenshot shows the ServiceNow configuration page for a Number field. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', 'Admin', and a search bar. The main header indicates the current configuration is for 'Number - MFE'. Below this, the configuration details are as follows:

- \* Table:** family\_expense
- Prefix:** MFE
- \* Number:** 1,000
- Application:** Global
- Number of digits:** 7

At the bottom of the configuration section, there are buttons for 'Update' and 'Delete', and a link for 'Show Counter'.

## Related Lists Configuration

### Family Expenses Form Related Lists

#### Configuration Interface:

- **Available Lists:** Attachments
- **Selected Lists:** Daily expenses
- **View name:** Default view

The related list displays all daily expense records associated with the current family expense record, providing a comprehensive view of individual transactions that contribute to the total family expense amount.

## Technical Implementation

### Database Schema

#### Primary Keys:

- Both tables use ServiceNow's standard Sys ID (GUID) as primary keys
- Number fields serve as user-friendly identifiers

#### Relationships:

- One-to-Many relationship between Family expenses and Daily expenses
- Foreign key relationship maintained through reference fields

#### Data Types:

- **String:** Text fields, comments, references

Relationship
Daily expenses
Update
Delete

Name
Application
Global ⓘ

Advanced ☐
Applies to table
Family expenses [u\_family\_expenses]

Queries from table
Daily expenses [u\_daily\_expenses]

This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see [the documentation](#). See also the article about the [recommended form of the script](#).

Query with
☒ Turn on ECMAScript 2021 (ES12) mode ⓘ

```

1  (function refineQuery(current, parent) {
2
3
4  // Add your code here, such as current.addQuery(field, value);
5
6  current.addQuery('u_date', parent.u_date);
7
8  current.query();
9
10
11 }) (current, parent);
12

```

Update
Delete

Related Links
[Run Point Scan](#)

- **Integer:** Numeric amounts and counters
- **Date:** Date fields for temporal tracking
- **Date/Time:** System audit timestamps

### Automation Features

1. **Automatic Number Generation:** Sequential numbering for both tables
2. **Real-time Aggregation:** Business rules update totals immediately
3. **Dynamic Detail Concatenation:** Expense details automatically formatted
4. **Audit Trail:** Complete tracking of all changes and updates

### Performance Considerations

- Indexed fields on date columns for efficient querying
- Business rule optimization for minimal database calls
- Efficient relationship queries using proper filtering

---

## System Benefits

### For Users

- **Simplified Data Entry:** Easy expense tracking interface
- **Automatic Calculations:** No manual total calculations required
- **Comprehensive Reporting:** Complete expense visibility
- **Audit Capabilities:** Full transaction history

### For Administrators

- **Configurable Forms:** Flexible layout options
- **Automated Workflows:** Reduced manual processing

- **Data Integrity:** Business rules ensure consistency
  - **Scalable Architecture:** Easy to extend and modify
- 

## Maintenance and Support

### Regular Maintenance Tasks

1. Monitor business rule performance
2. Review and optimize database queries
3. Validate data integrity periodically
4. Update number sequences as needed

### Troubleshooting Common Issues

- **Missing Aggregations:** Check business rule execution
  - **Number Generation Failures:** Verify number configuration
  - **Relationship Issues:** Confirm reference field mappings
  - **Form Display Problems:** Review form layout settings
- 

## Conclusion

The ServiceNow Family Expenses Management System provides a robust, automated solution for tracking family expenditures. With its comprehensive business rules, efficient relationships, and user-friendly forms, the system ensures accurate expense management while minimizing manual effort.

The implementation demonstrates best practices in ServiceNow development, including proper table design, effective business rule implementation, and intuitive user interface configuration.