# SSIS For Loop Container - Beginner to Advanced Tutorial

The For Loop Container in SSIS allows you to repeat tasks while a condition is true, just like a for loop in programming languages. This tutorial explains its properties, when to use it, and provides real-world examples with steps using Visual Studio 2022.

## What is the For Loop Container in SSIS?

The For Loop Container is a Control Flow container that repeats its inner tasks while a condition evaluates to true. It uses three main expressions:

• InitExpression – runs once before the first iteration.

• EvalExpression – checked before each iteration.

• AssignExpression – runs after each iteration.

## When & Where to Use

You should use a For Loop Container when:

1) You need to repeat a process a fixed number of times.

2) You are processing batches based on counters.

3) You want retry logic with a maximum limit.

4) You are iterating over months, days, or numeric ranges.

5) You are paginating through an API using page numbers.

## Common Variables

• Int32 counters: User::Counter, User::RetryCount

• DateTime: User::StartDate, User::CurrentDate

• String: User::SqlText, User::Status

• Boolean flags: User::HasMorePages, User::Succeeded

## Steps to Add a For Loop Container in VS 2022

1. Open Visual Studio 2022 with SSIS extension.

2. Create/Open an Integration Services Project.

3. In Control Flow, drag For Loop Container onto canvas.

4. Add required variables in the Variables window.

5. Double-click the For Loop Container and set InitExpression, EvalExpression, AssignExpression.

6. Place tasks inside the container.

## Real-World Examples

### Example 1 — Retry with Backoff

Scenario: Retry a failing Data Flow up to 3 times with a 10-second delay.

Variables: User::Retry, User::MaxRetry, User::DelaySeconds, User::LastAttemptSucceeded.

Loop settings: Init: @Retry=1, Eval: @Retry<=@MaxRetry && @LastAttemptSucceeded==FALSE, Assign: @Retry=@Retry+1.

Inside loop: Data Flow Task, Script Task to set success flag, Wait Task to delay between retries.

### Example 2 — Monthly Loads

Scenario: Loop from Month=1 to 12 to load monthly data.

Variables: User::Year, User::Month, User::SqlText.

Inside: Execute SQL Task with parameter mapping for Year and Month.

### Example 3 — API Pagination

Scenario: Loop through API pages 1 to N using a PageNo variable.

Variables: User::PageNo, User::TotalPages, User::EndpointBase, User::Url.

Inside: Build URL using expression, call API, load data.

### Example 4 — Batch Processing

Scenario: Process large table in batches of 50k rows.

Variables: User::Batch, User::BatchSize, User::TotalRows, User::Offset.

Inside: Data Flow with OFFSET/FETCH query using parameters.

### Example 5 — Backfill Last 30 Days

Scenario: Load daily data for the last 30 days.

Variables: User::DayOffset, User::DaysToLoad, User::LoadDate.

Inside: Calculate LoadDate using expression, pass to SQL Task.

## Common Pitfalls

• EvalExpression never becomes false → check AssignExpression.

• Type mismatch → match variable types and cast as needed.

• Expressions not updating → set in For Loop properties, not variable values.

• Incorrect parameter mapping in SQL Tasks.

## Hands-On Mini Labs

Lab A: Simple Counter Loop — Log iteration numbers using Script Task.

Lab B: Retry Data Flow — Retry failing task up to 3 times.

Lab C: Monthly Loader — Call stored procedure for each month.