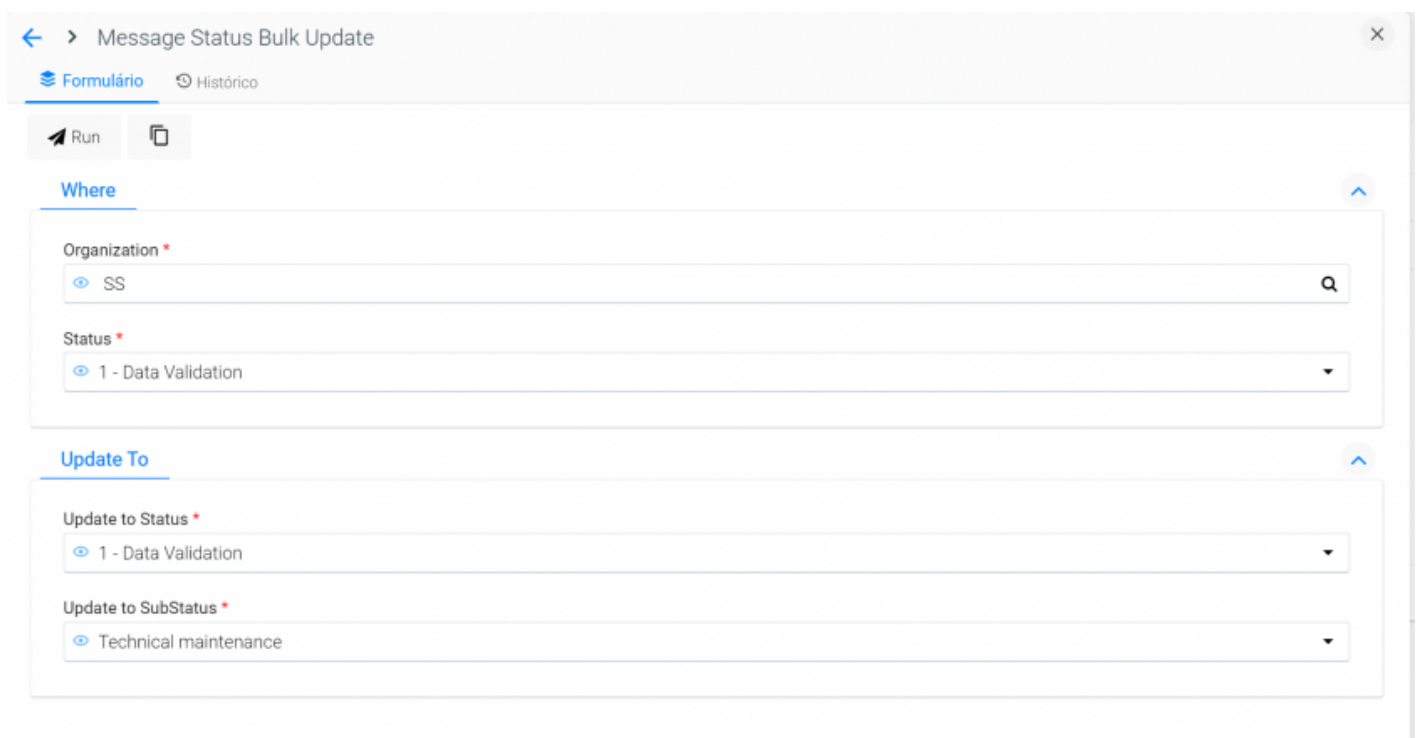


Bulk Update - Atualização de dados em Massa

Com o exemplo abaixo é possível atualizar varios dados de um ou mais forms com apenas 1 clique.

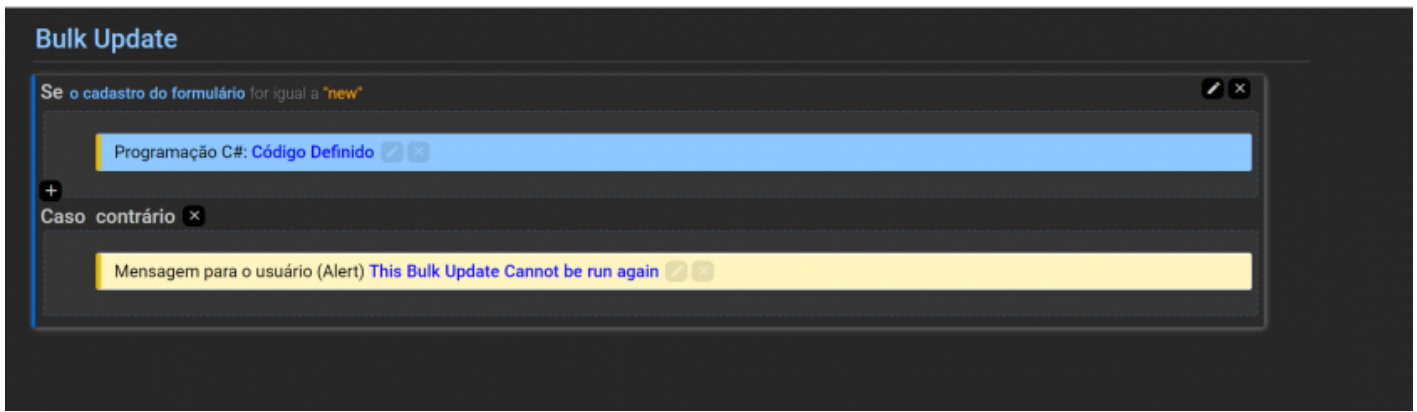
Criar um form com as informações que estarão no **Where** do Bulk Update e também os campos que deverão ser atualizados.

Exemplo:



The screenshot shows a web interface for 'Message Status Bulk Update'. At the top, there are navigation tabs for 'Formulário' (selected) and 'Histórico'. Below the tabs are 'Run' and 'Copy' buttons. The form is divided into two main sections: 'Where' and 'Update To'. The 'Where' section contains two fields: 'Organization' with a search input containing 'SS' and a search icon, and 'Status' with a dropdown menu showing '1 - Data Validation'. The 'Update To' section contains two dropdown menus: 'Update to Status' showing '1 - Data Validation' and 'Update to SubStatus' showing 'Technical maintenance'.

Criar uma regra de negócio ao "Salvar" da seguinte forma:



Abaixo está o código C# para colocar no item de programação da Regra de Negócio(acima) responsável por executar o Bulk Update (Atualização em massa):

IMPORTANTÍSSIMO: os scripts do SqlScript, atualizam o registro na RAIZ, isso é, **direto no banco de dados**, sendo assim, não executa nenhuma trigger, regra de negocio, controle de alteração, log de alteração etc.. que esteja vinculado(a) aos dados que estão sendo atualizados. **TENHA CUIDADO.**

ATENÇÃO: as informações dos formulários são sempre gravados em 2 tabelas. (1) tbl_formulario e (2) a tabela "oficial" de cada estrutura de formulário, então é OBRIGATÓRIO, atualizar os dados nas 2 estruturas conforme exemplo abaixo. Na tbl_formulario é criado um JSON com todos os dados, então o que vc precisa atualizar é a propriedade dentro do json do formulario que fica na coluna tbl_formulario **frm_json (o exemplo abaixo mostra como fazer essa atualização)**

```
public async Task ExecuteAsync()
{
    var where_status = FormContext.GetValue("where_status");
    var organization = FormContext.GetValue("organization");
    var update_to_status = FormContext.GetValue("update_to_status");
    var update_to_substatus = FormContext.GetValue("update_to_substatus");

    /*
    * O AgilityFlow grava todas as informações em JSON na tabela TBL_FORMULARIO
    * e também grava na tabela única da entidade.
    * Para fazer o bulk update, é necessário atualizar as 2 tabelas.
    * Importante também registrar que esse procedimento não executará nenhuma regra de negócio da
```

aplicação.

```
*/

// SQL update para atualizar o JSON na tbl_formulario
var sql_tbl_form = @"
    UPDATE tbl_formulario
    SET frm_json = jsonb_set(
        jsonb_set(frm_json::jsonb, '{{status}}', to_jsonb(@update_to_status)),
        '{{sub_status}}', to_jsonb(@update_to_substatus)
    )
    WHERE frm_id IN (
        SELECT id FROM x_tbl_message_queue
        WHERE organization = @organization
        AND status = @where_status
    );";

// SQL update para atualizar na tabela única da entidade
var sql_tbl_message = @"
    UPDATE x_tbl_message_queue
    SET status = @update_to_status, sub_status = @update_to_substatus
    WHERE organization = @organization
    AND status = @where_status;";

var sql = $"{sql_tbl_form} {sql_tbl_message}";

// Criando os parâmetros explicitamente
var parameters = new DbParameter[]
{
    new NpgsqlParameter("@update_to_status", update_to_status),
    new NpgsqlParameter("@update_to_substatus", update_to_substatus),
    new NpgsqlParameter("@organization", organization),
    new NpgsqlParameter("@where_status", where_status)
};

await FormContext.SqlScript.ExecuteAsync(sql, parameters);
}
```

Revision #4

Created 15 March 2024 18:42:20 by agilityflow

Updated 9 February 2025 14:04:02 by agilityflow