Cloudera Flow Management 2.1.5

Using parameter context inheritance

Date published: 2019-06-26
Date modified: 2022-12-13

https://docs.cloudera.com/

Legal Notice

Contents

What is parameter context inheritance?

With inherited parameter contexts, you can make granular updates to parameter contexts and have the changes propagate to all contexts that inherit from them.

Parameter context is a named set of parameters with applied access policies in Apache NiFi. Using parameter contexts is a powerful way to make flows more portable. Paired with process groups, parameter contexts allow different values to be supplied to the same basic flow in multiple NiFi instances, or even within the same overall NiFi flow in a single instance. However, as the number of parameters grows, parameter contexts can become more difficult to maintain and can result in a monolithic parameter context. If you collect all parameters in a root parameter context, you are parameterizing your process groups with more parameters than they need, as some of the parameters included in the parameter context are applicable only to certain process groups.

The goal is to avoid parameter duplication and yet still compose groups of parameters as needed. Starting with Apache NiFi 1.15.0, parameter contexts support inheritance, so you can add inherited parameter contexts to an existing parameter context. This feature brings more flexibility and maintainability to NiFi's already powerful parameter context framework. Using inheritance makes it easier to manage common sets of parameters used across many flows and adjust specific parameters for a given instance as needed. Creating parameter contexts through inheritance provides the opportunity to use a structure of smaller, logically grouped parameter contexts and design your flows with these smaller, composable contexts. Once you set up the inheritance hierarchy, you can make granular updates and have the changes propagate to all other contexts that inherit from it.

For more information about process groups and parameter contexts, see the *Apache NiFi User Guide*.

Related Information

Configuring a Process Group, Apache NiFi User Guide

Parameter Contexts, Apache NiFi User Guide

Example for configuring parameter context inheritance

In this example you can see how parameter context inheritance is configured for two existing process groups (Source ABC Kafka to S3 and Source DEF Kafka to GCS) to avoid repeating the Kafka brokers or the common site-specific parameters, and yet still composing groups of parameters as needed. The small, composable parameter contexts are logically grouped through inheritance.

Creating the basic parameter contexts

Follow these steps to set up your individual parameter contexts and add the relevant parameters to them.

Procedure

1. Create a parameter context for the common Kafka settings:
   a) Select Parameter Contexts from the top-right Global menu in the NiFi UI.

      The NiFi Parameter Contexts dialog opens.
   b) Click the (+) button in the top-right corner of the NiFi Parameter Contexts dialog to add a new parameter context.
   c) Add a name for your parameter context on the SETTINGS tab.

      In this example the name is Kafka.
   d) Switch to the PARAMETERS tab.
   e) Click the (+) button to add the parameters you need for configuring your data flow.
   f) Provide a name and a value in the Add Parameter dialog. You can also add a description of the parameter, if you want to.

      In this example, the following parameters are added:

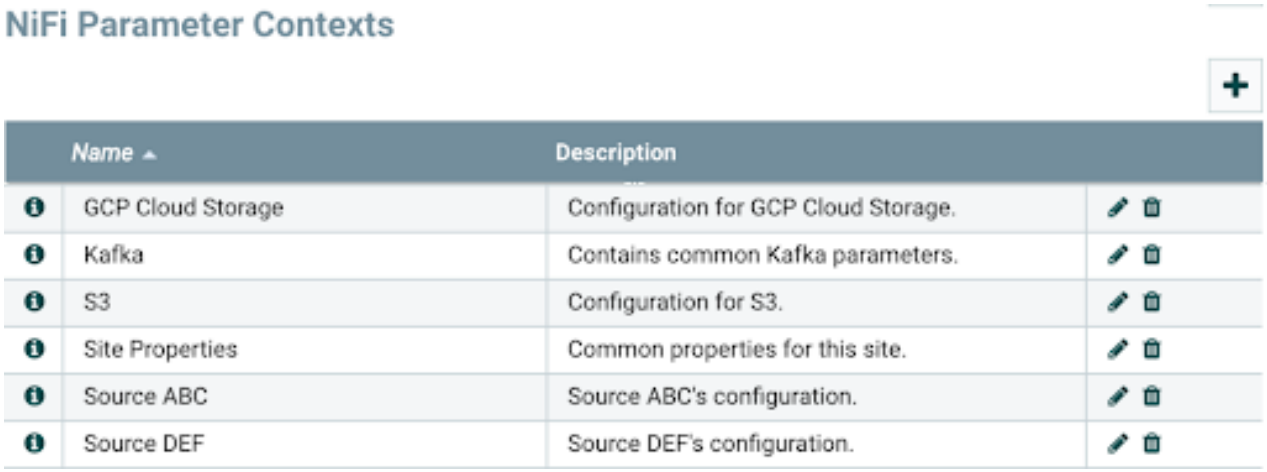
      - Name: Kafka Brokers / Value: localhost:9092
      - Name: Kafka Group ID / Value: MyGroup

   g) Click APPLY to save the parameter context.
2. Create the remaining parameter contexts by repeating sub-steps a to g outlined in Step 1, using the following names and values:

Parameter context	Parameter name	Parameter value
GCP Cloud Storage	GCP Project ID	my-project
	GCP Bucket	my-bucket
S3	S3 Bucket	my-bucket
	S3 Region	us-west-2
Site Properties	Site Identifier	1234
	Site Data Manager	MyDM
Source ABC	Kafka Topic	source-abc-topic
Source DEF	Kafka Topic	source-def-topic

Results

You are ready with the individual parameter contexts that you will use to create your inheritance hierarchy. Your parameter contexts should look like this:



Setting up parameter context inheritance

Follow these steps to configure inherited parameter contexts.

About this task

Now that you have the individual parameter contexts set up, you can begin grouping them together, and create higher-level parameter contexts to build out your inheritance hierarchy and use them with the two process groups (Source ABC Kafka to S3 and Source DEF Kafka to GCS). The goal is to have one parameter context for each process group, composed of smaller units.

In this example, site properties and Kafka parameters are needed for both process groups. Additionally, each process group will have a source-related and a destination-related parameter context.

Before you begin

You have created two process groups in Apache NiFi:

- Source ABC Kafka to S3 that needs parameters for your Kafka brokers, Source ABC's topic, and for the S3 bucket
- Source DEF Kafka to GCS that needs parameters for the same Kafka brokers, Source DEF's topic, and for the GCP bucket
- In both Process Groups, you would like to use some common parameters tagging flowfiles with site-specific information

Procedure

1. Design the inheritance logic.

   In this example, the following structure is used:

   - Kafka and Site - inherits from Kafka and Site Properties
   - ABC Kafka and Site - inherits from Kafka and Site and Source ABC
   - DEF Kafka and Site - inherits from Kafka and Site and Source DEF
   - ABC Kafka to S3 - inherits from ABC Kafka and Site and S3
   - DEF Kafka to GCS - inherits from DEF Kafka and Site and GCP Cloud Storage

   Based on the above structure, the hierarchy for the two top-level parameter contexts looks like this:

   ABC Kafka to S3 parameter context

   - ABC Kafka and Site parameter context
     - Kafka and Site parameter context
       - Kafka parameter context
         - Kafka Brokers parameter with value localhost:9092
         - Kafka Group ID parameter with value MyGroup
       - Site Properties parameter context
         - Site Identifier parameter with value 1234
         - Site Data Manager parameter with value MyDM
     - Source ABC parameter context
       - Kafka Topic parameter with value source-abc-topic
   - S3 parameter context
     - S3 Bucket parameter with value my-bucket
     - S3 Region parameter with value us-west-2

   DEF Kafka to GCS parameter context

   - DEF Kafka and Site parameter context
     - Kafka and Site parameter context
       - Kafka parameter context
         - Kafka Brokers parameter with value localhost:9092
         - Kafka Group ID parameter with value MyGroup
       - Site Properties parameter context
         - Site Identifier parameter with value 1234
         - Site Data Manager parameter with value MyDM
     - Source DEF parameter context
       - Kafka Topic parameter with value source-def-topic
   - GCP Cloud Storage parameter context
     - GCP Project ID parameter with value my-project
     - GCP Bucket parameter with value my-bucket

2. Configure parameter context inheritance in NiFi for ABC Kafka to S3.
   a) Create a new parameter context named Kafka and Site and go to the INHERITANCE tab.

      You can see all of the individual parameter contexts.
   b) Drag Kafka and Site Properties to the right under Selected Parameter Context.
   c) Click APPLY, and then edit the Kafka and Site Parameter Context to see the new parameters.

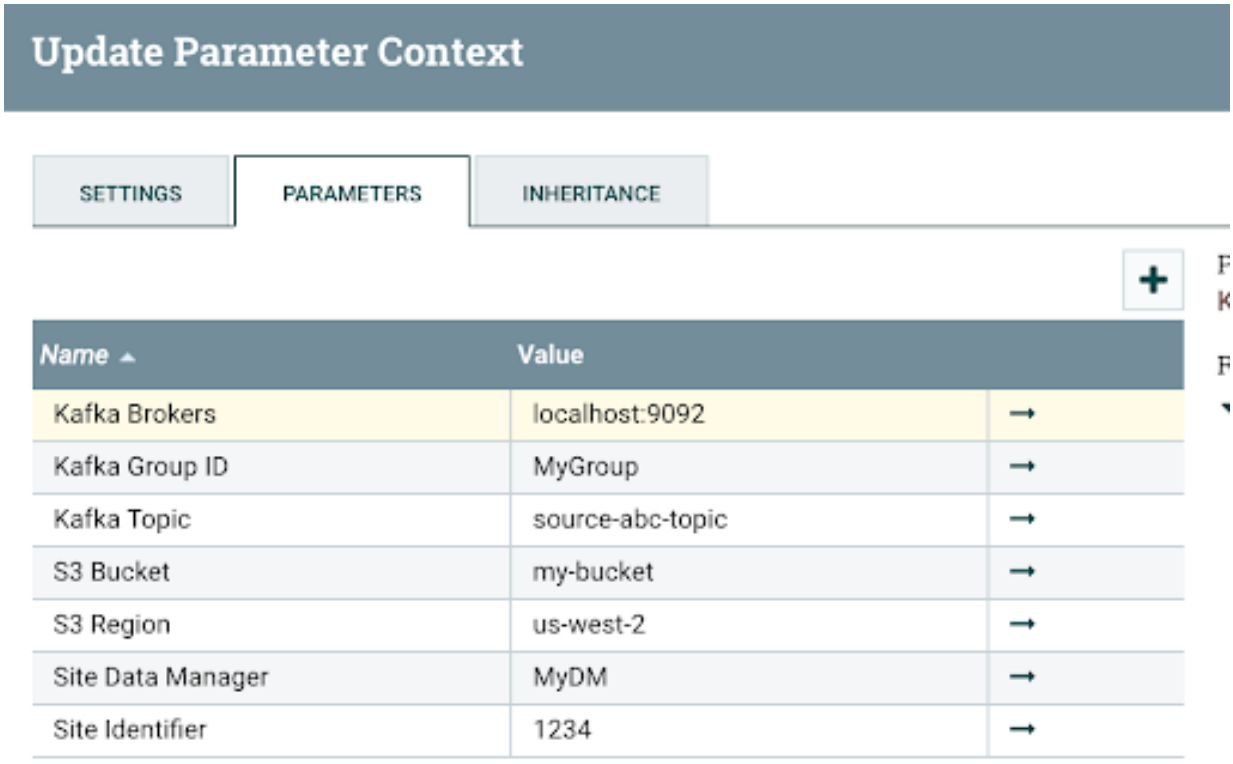
      All four parameters from the two parameter contexts are now available in the view, and each of them has an (arrow) icon. You can click one of these to navigate to the parameter context in which the parameter is defined and edit its value.

      Note:

      You can drag/drop from the Available Parameter Contexts to the required inherited order in the Selected Parameter Context column.

      When you remove a parameter context from the Selected Parameter Context list, it returns to the Available Parameter Contexts that is sorted alphabetically.
   d) Repeat steps 1-3 to create ABC Kafka and Site Parameter Context inheriting all parameters from the Kafka and Site and the Source ABC Parameter Contexts.
   e) Finally, repeat steps 1-3 to create ABC Kafka to S3 Parameter Context inheriting all parameters from the ABC Kafka and Site and the S3 Parameter Contexts.

   This is how ABC Kafka to S3 looks like after the full setup:



3. Configure parameter context inheritance for DEF Kafka to GCS.

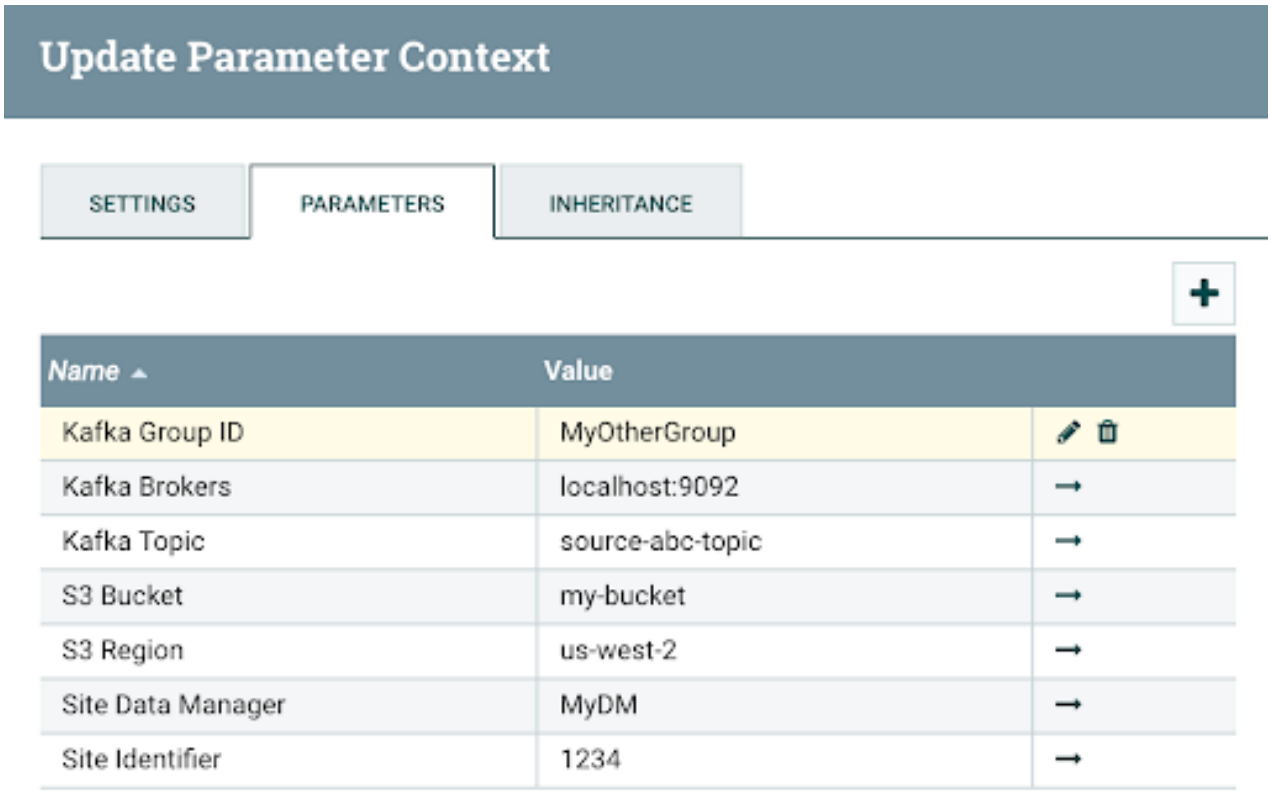
   Follow the steps you completed when creating the ABC Kafka to S3 Parameter Context.

Parameter overriding

When a parameter context inherits parameters from other contexts, you can still manually override parameters with a different value, if needed.

In this example, another process group is added that uses the same parameters as ABC Kafka to S3, but with a Kafka Group ID of MyOtherGroup.

To configure it, you can override the original parameter value by creating a new parameter context that inherits from ABC Kafka to S3 and adding a Kafka Group ID parameter with the MyOtherGroup value directly to that new context:



In this case the Kafka Group ID with its value MyOtherGroup appears at the top of the parameter list, and not where it should be in alphabetical order. Direct parameters always appear at the top of the list so you can see them grouped together.

You can edit the manually set parameter directly from this parameter context. So in this case, you are editing the parameter in the new context and not the inherited parameter.

Parameter overriding order

- Direct parameters always take precedence.
- Parameters inside directly inherited parameter contexts take precedence, from top to bottom in the Selected Parameter Context column of the INHERITANCE tab.
- This recursively repeats in a depth-first manner for as many layers of inherited parameter contexts exist.