

Flat File Records Without Data



This page explains the **Flat File Adapter** in Bridge context. If you were looking for the same information regarding the **PAS Designer**, refer to **Flat File Adapter** in the Designer guide.

Sometimes, flat files contain records without any data but just the field separators, like in line 4 to 7 of this example below:

Figure: Flat File Records without Data

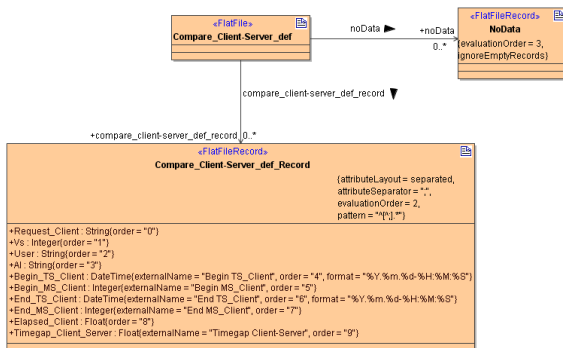
```
0 10 20 30 40 50 60 70
1 Request_Client;Vs;User;AI;Begin_TS_Client;Begin_MS_Client;End_TS_Client;End_MS_Client;
2 OORTI;1;SZH229;23;2005.11.07-08:30:26;604000;2005.11.07-08:30:27;72000;0.468;
3 QRYCODEITEM;10;SZH229;23;2005.11.07-08:30:27;72000;2005.11.07-08:30:30;94400C
4 ;;;;;;;;;;;;;;
5 ;;;;;;;;;;;;;;
6 ;;;;;;;;;;;;;;
7 ;;;;;;;;;;;;;;
8 QRYCODEITEM;10;SZH229;23;2005.11.07-08:30:31;22000;2005.11.07-08:30:33;64400C
9 OORTI;1;SZH229;23;2005.11.07-08:30:33;691000;2005.11.07-08:30:35;939000;2.24E
10 QRYCURRENCY;10;SZH229;23;2005.11.07-08:30:35;939000;2005.11.07-08:30:38;4370C
```

There are two ways to skip the records containing no data.

Handle Empty Records With a Pattern

Specify a pattern in tagged value **pattern** that only matches with the records having data. In our example, the regular expression `^[^;]*` supposes that a record is without data when it starts with a separator.

Figure: Flat File – Exclude Records Without Data



The flat file parser will abort parsing when a record does not match any `<<FlatFileRecord>>` class. Thus we have to specify a `<<FlatFileRecord>>` class for the records without data as well.

Class **NoData** has no pattern. Actually it matches all records, including the records containing data. Thus, it is important to set tagged value **evaluationOrder** to ensure that this record type will be parsed last on parsing the records.

NoData has no attributes defined, therefore no fields are parsed. Thus, the object of type **noData** would be created without having any content. To avoid such empty records, set tagged value **ignoreEmptyRecords=true** on **NoData** class.

Use Tagged Values on the <<FlatFileRecord>> Class

Use a combination of tagged values **ignoreEmptyStrings** and **ignoreEmptyRecords** on the `<<FlatFileRecord>>` class.

Figure: Flat File – Ignore Empty Records and Strings

On this Page:

- [Handle Empty Records With a Pattern](#)
- [Use Tagged Values on the <<FlatFileRecord>> Class](#)

Related Pages:

- [Flat File Adapter Reference](#)

Other Common Flat File Issues:

- [Flat File Records Without Data](#)
- [Flat File With Header Lines](#)
- [Handling of Empty Lines When Composing a Flat File](#)
- [Handling of White Spaces When Parsing Flat Files With Fixed Length](#)
- [Mapping of Hierarchical Record Structures](#)
- [Using Patterns to Separate Attributes](#)

«FlatFileRecord» Compare_Client-Server_def_Record	
	{attributeLayout = separated, attributeSeparator = ";", evaluationOrder = 2, ignoreEmptyRecords, ignoreEmptyStrings}
+Request_Client : String(order = "0")	
+Vs : Integer(order = "1")	
+User : String(order = "2")	
+AI : String(order = "3")	
+Begin_TS_Client : DateTime(externalName = "Begin TS_Client", order = "4", format = "%Y.%m.%d-%H:%M:%S")	
+Begin_MS_Client : Integer(externalName = "Begin MS_Client", order = "5")	
+End_TS_Client : DateTime(externalName = "End TS_Client", order = "6", format = "%Y.%m.%d-%H:%M:%S")	
+End_MS_Client : Integer(externalName = "End MS_Client", order = "7")	
+Elapsed_Client : Float(order = "8")	
+Timegap_Client_Server : Float(externalName = "Timegap Client-Server", order = "9")	

Ignoring empty strings will lead to the Flat File Adapter processing empty strings to `NULL`, and ignoring empty records will lead to the Flat File Adapter not processing a record, if it has no data.

Note, that a record containing only empty **Strings** is not empty – in opposition to a record composed from `NULLS`.