

User Guide for Consistency Checker

Ericsson Dynamic Activation

USER GUIDE



Contents

1	Instruction	1
1.1	Target Group	1
1.2	Prerequisites	1
2	Getting Started	3
2.1	Logging In	3
2.2	Launch Page	3
2.3	A Basic Analysis Order	4
3	Use Cases	7
3.1	Recurrent Analysis with Data Extraction	7
3.2	Two-step Analysis	8
4	Working with Analysis Orders	11
4.1	Schedule Analysis Order	11
4.2	Specify Data Sources	12
4.3	Select Analysis Type	16
4.4	Define Rule Specification	16
5	Reading Analysis Report	19
5.1	Read a Report	19
5.2	Read a Definition	20
5.3	Read a Statistics	21
6	Appendix	23
6.1	Comparison Rules	23
6.2	Rule Conditions	25
	Reference List	27





1 Instruction

This document is a guide to the Graphical User Interface (GUI) of the Consistency Checker in Ericsson™ Dynamic Activation.

1.1 Target Group

The target group for this document is the Consistency Checker users.

1.2 Prerequisites

- For Operation & Management usage, ensure that the following conditions are met:
 - Consistency Checker is installed successfully. Refer to *Installation Instruction for Consistency Checker on Glassfish Server Open Source Edition*, Reference [1].
 - Users understand the basic concept of consistency analysis. Refer to *Function Specification Consistency Checker*, Reference [2].
- For troubleshooting and maintenance usage, refer to *System Administrators Guide for Consistency Checker*, Reference [3].
- For advantage usage, for example, customized consistency checking, refer to *Programmers Guide for Consistency Checker*, Reference [4].



2 Getting Started

This section is designed to help users get started using Consistency Checker as soon as possible.

2.1 Logging In

1. Use a web browser and direct it to the Consistency Checker web management address:

`http://<host>:<port>/cc_management`

For example:

`http://localhost:8080/cc_management`

2. Enter the user name and password provided by the system administrator.

2.2 Launch Page

Figure 1 shows the launch page of Consistency Checker after logging in successfully.

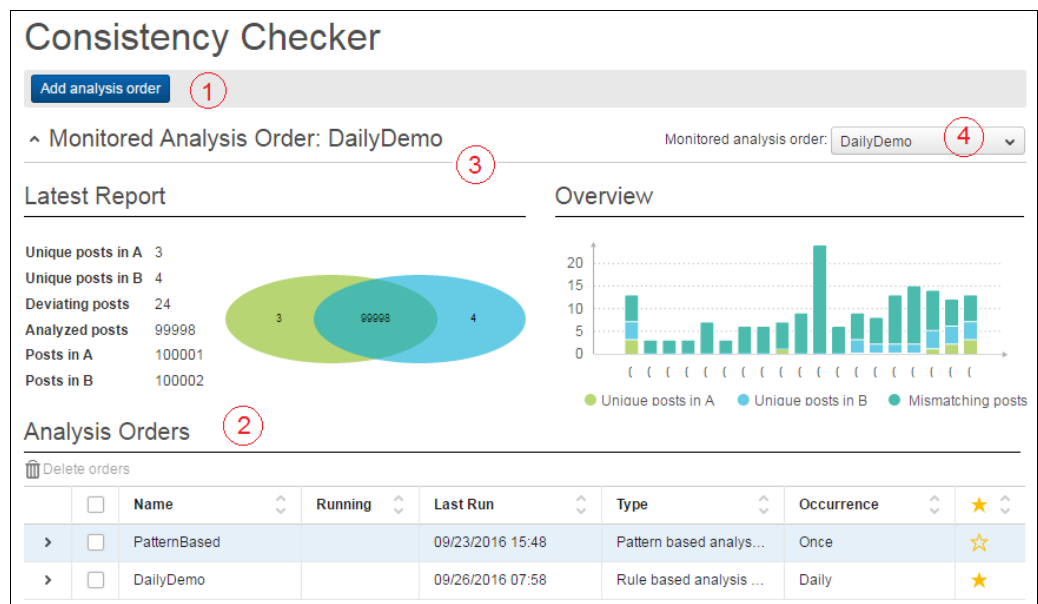




Figure 1 Launch Page

In the launch page:

- 1 Click **Add analysis order** to add an order of consistency analysis.



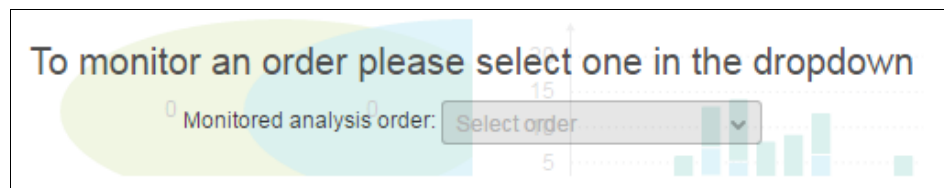
2 Here lists all added analysis orders. Users can:

- Hover the mouse over analysis orders, and select “favorite” ones by using ★.
- Click ➤ to view the analysis order execution report.
- Click  to sort the analysis orders ascendingly or descendingly.
- Use ☐ and then  Delete orders to delete one or more analysis orders.

Note: When deleting an analysis order, ensure that the order is not depended by other orders.

3 Here displays a summary report of a monitored analysis order.

- If the GUI displays the following image, users can select a “favorite” order (★) in area 2.



- When a monitored order is selected, GUI displays a summarized report of the latest execution.
- Users can click ▼ (or ▲) to expand (or collapse) the summarized report.

4 This drop-down list is visible when there is a monitored analysis order. Users can:

- Add (or remove) list items by selecting (or un-selecting) “favorite” orders (★) in area 2.
- Select another analysis order in the drop-down list.

2.3 A Basic Analysis Order

In the launch page, click **Add analysis order** to access the **Add Analysis Order** page, as shown in Figure 2.



Add Analysis Order

Analysis Details

Name ☒ OK

Description

Schedule ①

Data Sources

Data Source A	Data Source B
Type <input checked="" type="radio"/> Dump file <input type="radio"/> Extraction <input type="radio"/> Online	Type <input checked="" type="radio"/> Dump file <input type="radio"/> Extraction <input type="radio"/> Online
Source <input type="text" value="dsA.csv"/> <input type="button" value="..."/> ②	Source <input type="text" value="dsB.csv"/> <input type="button" value="..."/>

Analysis Type

Type ③

Your analysis will find inconsistencies based on identified data relation pattern

Figure 2 An Easy Analysis Order

In the example shown in Figure 2:

- 1 If not specify a date and time, the analysis order is executed immediately.
- 2 Specify two dump files for **Data source A** and **B**.

The two dump files must have a common identifier, and data are indexed based on the identifier.

- 3 Select the analysis type as **Pattern based analysis**, which means:
 - No need to define any rules for data analysis.
 - The Consistency Checker finds the rules beneath the data pattern, and then use those rules to analyze data.



When click **Save**, the order can be found in the launch page, and will be executed as scheduled.

For more information on how to interpret the analysis report of the order, see Section 5 on page 19.



3 Use Cases

This section provides example use cases for using Consistency Checker in best practices.

3.1 Recurrent Analysis with Data Extraction

This use case is to identify whether inconsistent data exist between two data sources, for example, an HLR and an SDP data sources.

Assumption

- Both data sources files are generated at 01:00:00 on a daily basis, but:
 - The HLR generates a valid dump file `myData.csv` in the `<CheckerHomeDir>/var/dump` directory. The latest one always overwrites the previous one.
 - The SDP generates a log file `mySDP_<timestamp>.log` (for example, `mySDP_2016-10-10.log`) in the `<CheckerHomeDir>` directory.
- The following are deployed in Consistency Checker during integration phase:
 - An extraction handler `mySDP_Extractor`, which is used to extract data from a `mySDP_<timestamp>.log` file for analysis
 - The data models of both data sources

Table 1 Analysis Order Example

Attributes	Recommended Settings	
Analysis Details		
Schedule	Daily, at 01:10:00	
Data Sources		
	Date Source A	Data Source B
Type	Dump file	Extraction
Extraction handler	N/A	mySDP_Extractor
Sources 1	myData.csv	mySDP_* Note: Consistency Checker always uses the latest file whose file name fits the pattern.



Attributes	Recommended Settings
Analysis Type: Rule based analysis order	
Rule Specification	
Data model A	myData_data_model
Data model B	SDP_data_model
Rules	According to business logic.

3.2 Two-step Analysis

The two-step process combines the advantages of both the offline and online analysis. In this use case, the analysis result can accurately reflect the real-time data consistency, and the impact on the network is minimal.

- **Step 1**, define an offline analysis order on dump files generated at different time points.

This is used to identify suspected inconsistencies on entire subscriber base, which includes:

- Real inconsistencies in network.
- False inconsistencies caused by dump files. This is because it can take long time to generate a dump file, and the oldest items can be different from the newest ones in such a file.

- **Step 2**, define an online analysis order based on the Step 1 order.

This is used to confirm the suspected inconsistencies found in **Step 1**, and rule out false ones.

Assumption

- A myData and SDP dump files are exported to the `<CheckerHomeDir>/var/dump` at 01:00:00 on every Monday. New files are always overwrite previous ones.
- The following are deployed in Consistency Checker during integration phase.
 - The data model of both data sources
 - The connection to real time data sources of myData and SDP



Table 2 Analysis Order Example

Attributes	Recommended Settings	
	Step 1 Analysis Order	Step 2 Analysis Order
Analysis Details		
Schedule	Weekly, at 01:00:00, Monday	Weekly, at 01:30:00, Monday Note: Reserve some time for executing Step 1 analysis order.
Data Sources		
Type	Dump file for both Data source A and B	Online
Sources 1	Data source A: myData.csv Data source B: mySDP.csv	Data source A: myData_database Data source B: mySDP_database
Scope	N/A	Previous analysis: Select the name of the Step 1 analysis order.
Analysis Type	Rule based analysis order	Rule based analysis order
Rule Specification		
Data model A	myData_data_model	One of the following: <ul style="list-style-type: none"> • Load from the template myData_SDP. • Define another rule specification according to business logic. Note: The rule specifications for Step 1 and 2 are not necessary to be the same.
Data model B	SDP_data_model	
Rules	<ul style="list-style-type: none"> • Create a rule specification according to business logic. • Save it as a rule template myData_SDP. 	





4 Working with Analysis Orders

A typical analysis order defines the following:

- **Analysis Details** – Schedule when to analyze
- **Data Source** – Specify what to analyze
- **Analysis Type** – Define how to analyze
 - If the analysis is based on rules, a **Rule Specification** must be defined.

Use case examples can be found in Section 3 on page 7.

4.1 Schedule Analysis Order

This section describes how to schedule an analysis order. Figure 3 shows the GUI.

The screenshot shows a web form titled "Analysis Details". It has three main sections:

- Name:** A text input field containing "myAnalysis" with a green checkmark and "OK" button to its right.
- Description:** A large text area containing the placeholder text "This analysis is used to ...".
- Schedule:** A section with a dropdown menu set to "Once" and a date/time input field showing "MM/DD/YYYY hh:mm:s" with a calendar icon to its right.

Figure 3 Define Analysis Details

When schedule an analysis, consider the following issues:



- (Optional) Use  to specify a date and time for execution. See Table 3.
If omitted, Consistency Checker uses the current system time.
- When schedule a two-step analysis order, it is important to synchronize the execution timing and interval.



Table 3 Calendar Settings

Reurrence	Date and time in  means
Once	When to execute the analysis.
Daily	Starting from the specified date, execute the analysis every day at the specified time.
Weekly/ Monthly	Starting from the specified date, execute the analysis on the same day of a week/month, at the specified time.

4.2 Specify Data Sources

This section describes how to specify **Data source A** and **B** to perform analysis on.

4.2.1 How to Chose Data Source Types

When adding an off-line analysis order:

- Use **Dump files** type when the data to be analyzed are stored in Comma Separated Values (CSV) files and are indexed based on an identifier.
- Use **Extraction** type when the data to be analyzed are stored in various data formats. In this case, an appropriate **Extraction handler** needs to be deployed to extract data from such files in integration phase.

When adding a real time analysis order:

- Use **Online** type for both **Data source A** and **B**.

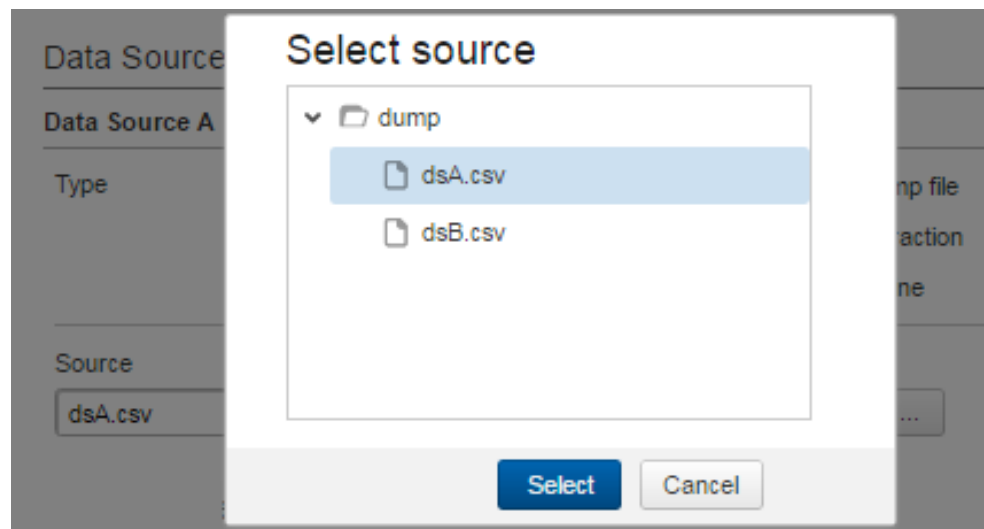
The online data sources need to be defined in integration phase.

4.2.2 Use Dump Files

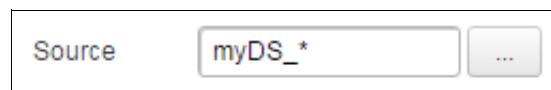
The **Dump files** type is used when the data to be analyzed are stored in `.csv` dump files and are indexed based on an identifier. Such dump files are stored in the `<CheckerHomeDir>/var/dump`. For instruction on how to change the default setting, refer to *Programmers Guide for Consistency Checker*, Reference [4].

When use **Dump file** type of data sources, users can:

- Select a particular dump file.



- Use wildcard “*” to specify a file name pattern.



Constancy Checker always uses the latest matching instance to execute the analysis order.

4.2.3 Use Data Extraction

In most common cases, the data to be analyzed are stored in various types of data files, and cannot be used directly by Consistency Checker. It is necessary to use an **Extraction handler** to extract the data to valid dump files.

By default, the data files are stored in `<CheckerHomeDir>`, and the extracted dump files are stored in `<CheckerHomeDir>/var/dump`. For instruction on how to change the default settings, refer to *Programmers Guide for Consistency Checker*, Reference [4].



The screenshot shows the 'Data Sources' configuration window. It is divided into two panels: 'Data Source A' and 'Data Source B'. Both panels have a 'Type' section with radio buttons for 'Dump file', 'Extraction' (selected), and 'Online'. Below this is an 'Extraction handler' dropdown menu. In 'Data Source A', the handler is 'HgSupDataParser' (annotated with a red circle 1). Below it is an '+ Add argument' button (annotated with a red circle 2). There are two 'Source' input fields: 'Source 1' with '1M_HgSup1.log' (annotated with a red circle 3) and 'Source 2' with '1M_HgSup2.log'. Each source field has a file selection icon (three dots) and a red 'x' icon. Below the sources is an '+ Add source' button. At the bottom, there is a 'Sequential extraction' checkbox which is checked (annotated with a red circle 4). 'Data Source B' has a similar layout but with a different handler 'CopyCSVFileDataSouc...' and a single source '1M_DemoNE3.csv'.

Figure 4 Use Data Extraction

When use **Extraction** type of data source, consider the following:

- 1 **Extraction handlers** listed in the drop-down list are deployed in integration phase.

For more information on how to develop and deploy a customized **Extraction handler**, refer to *Programmers Guide for Consistency Checker*, Reference [4].

- 2 If an **Extraction handler** requires argument(s) to execute extraction (for example, a filter to filter data), use **Add argument** to add them. Contact the **Extraction handler** developer for more information of the arguments.
- 3 Users can select a particular file to extract data for analysis, or use wildcard "*" to specify a file name pattern.

Consistency Checker always extracts the matching instance with the latest timestamp for an analysis order.

- 4 **Sequential extraction** is shown if both **Data source A** and **B** use **Extraction** type of data sources.

When **Sequential extraction** is selected, Consistency Checker extracts **Data source A** first and then **B**. This can make the long execution time shorter by not operating on the same I/O resources simultaneously for both extractions.



4.2.4

Use Online Data

Figure 5 Use Online Data

When using **Online** type of data sources, consider the following:

- 1 Online data sources listed in the drop-down list are prepared during integration phase. For instructions on how to prepare them, refer to *Programmers Guide for Consistency Checker*, Reference [4].
- 2 When **Number ranges** is selected, users need to define one or more ranges to limit the scope of the analysis order.
 - Use the identifier of the data source to define the range. For example, when the data source identifier is `MSISDN`, a specified number range from `46455277011` to `46455277399` means that the analysis will run only for the `MSISDN`s throughout that range.
 - The starting of a range must be lower than the ending.
- 3 When **Previous analysis** is selected, the current analysis order only analyzes the unique and deviating data reported in the selected previous analysis.

- The data sources of the previous and current analysis order must use the same identifier.



For example, if **MISDN** is used as an identifier in a previous off-line analysis order, the identifier of the current online analysis order must represent the same thing.

- If the previous analysis is a recurrent one, the latest successful previous analysis is used.
- If the previous analysis failed due to any reason, the current analysis fails too.

4.3 Select Analysis Type

The general guideline of selecting an analysis type is to:

- Select **Rule based analysis order** when:
 - Users are aware of how the data sources should relate to each other.
 - Complex rules or rule conditions need to be applied to the analysis.

In this case, a rule specification that is used to evaluate the data consistency must be defined. See Section 4.4 on page 16.

- Select **Pattern based analysis order** when the data relation between two data sources is unclear.

In this case, the Consistency Checker finds rules based on the identified data relation pattern, and uses those rules to evaluate the data consistency.

4.4 Define Rule Specification

This section is only available for **Rule based analysis**.

Consistency Checker uses a rule specification to analyze both **Data source A** and **B**.

A rule specification must contain one identifier rule, and none, one, or more other rules. Data pairs that:

- Violate the identifier rule – are considered as “Unique posts”.
- Violate any non-identifier rule(s) – are considered as “Mismatching posts”.



4.4.1

Add or Edit Rules

Rule Specification

Data model A dsA

- City
- FamilyName
- IMSI
- PhoneNumber
- PrivateID
- State
- Street
- Surname
- ZIPcode

Data model B dsB

- Mail
- PrivateID
- PublicID
- SIPURI
- State

Identifier ☐
Rule B contains A
Conditions A.IMSI = EXIST
Description

Add

Model A (dsA)	Model B (dsB)	Rule	Identifier	
PrivateID	PrivateID	Equal	true	

Save as template **Load template**

Figure 6 Add or Edit Rules

When adding or editing rules for an analysis order, consider the following:

- 1 The selected **Data models** must reflect the structure of the **Data sources A** and **B** respectively.
- 2 Click an item in **Data model A** and then another item in **B**. This shows areas **3–5**.
- 3 Select the **Identifier** checkbox for a rule that represent the identifier pair of both data models.
- 4 **Rule** is to define a comparison rule to analyze the data sources. See Section 6.1 on page 23 for description of the default rule types.
- 5 **Conditions** is to edit pre-conditions that determines if a rule shall be evaluated or not. See Section 6.2 on page 25 for description of the rule condition types.

Note: If the condition is not fulfilled, the rule will be ignored.

- 6 After clicking **Add**, the added rule is listed in the table. Users can select one rule and then edit or delete it.

- 7 Common used rule specifications can be saved as a template for later use. See Section 4.4.2 on page 18.

Note: The **Data model** and comparison rules listed in the drop-down lists are prepared during integration phase. For more information on how to develop and deploy customized ones, refer to *Programmers Guide for Consistency Checker*, Reference [4].

4.4.2 Use Rule Templates

When a rule specification is saved as a rule template, the same set of rules can be reused for other analysis orders by using the **Load template** function.

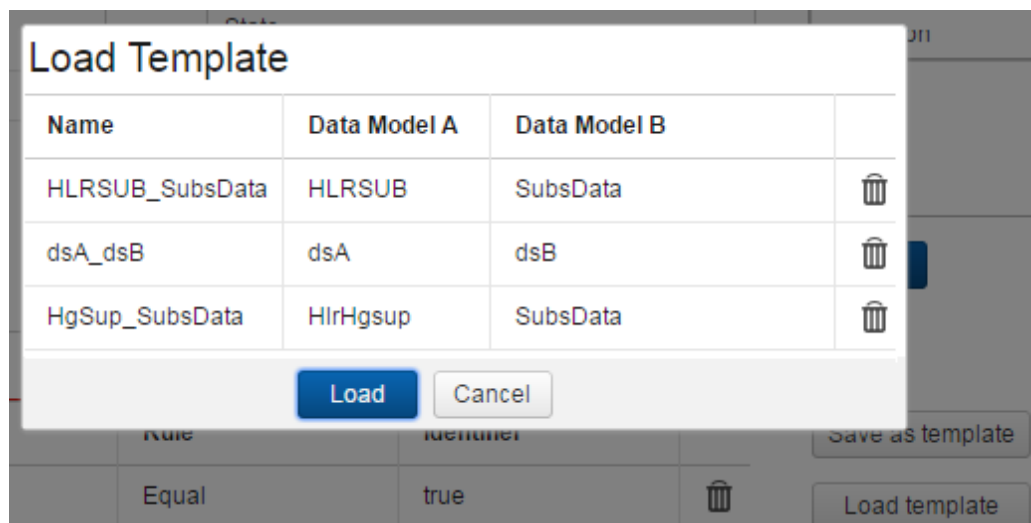


Figure 7 Use Rule Templates

Loading a rule template is to create a rule specification instance for an analysis order, which means:

- If any rules were added manually, they are removed after loading the template.
- Users can edit the rule specification, such as updating, adding, or removing rules. Any changes in the specification are not applied to the rule template.
- Removing a rule template does not impact any analysis orders that used the template previously.



5 Reading Analysis Report

This section describes how to understand the report and result of an analysis execution.

5.1 Read a Report

Figure 8 shows an example of an analysis execution report.

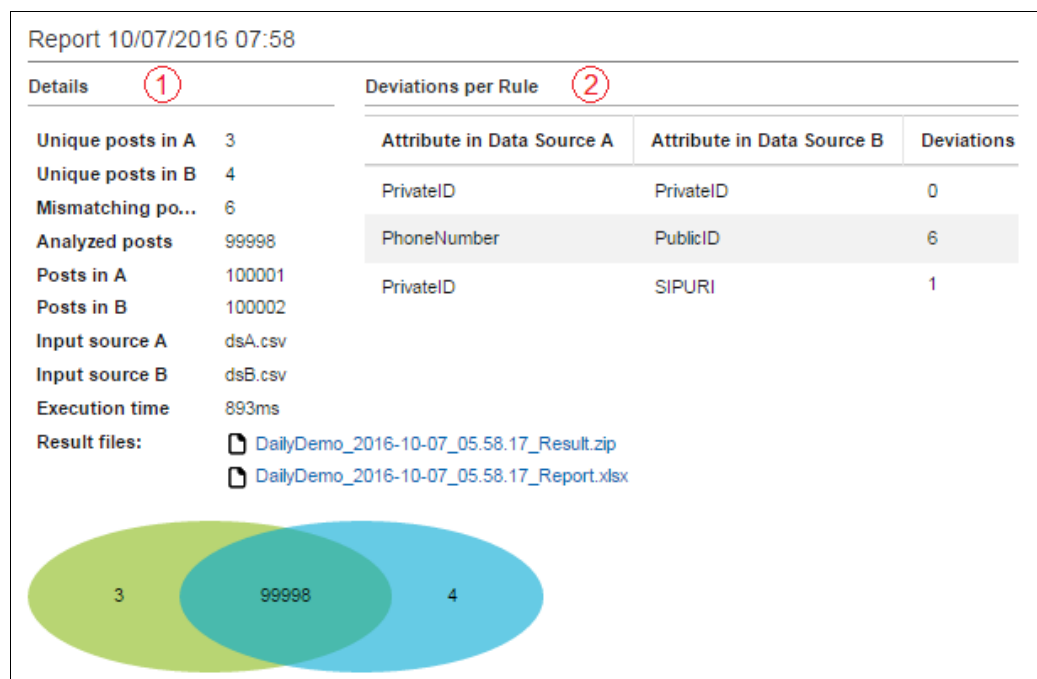


Figure 8 Report Example

Tips for reading a **Report**:

- Be aware of that, the **Mismatching posts** number in area 1 can be less than the sum of the **Deviations** column in area 2. That is because one post can violate several rules, which results in multiple deviations.
- Click the link to download the result .zip file, which includes all details for further post processing of the result.

The file format is described in *Programmers Guide for Consistency Checker*, Reference [4].

- Click the link to download the Excel report .xlsx file, which includes human readable details. Figure 9 shows an example.



	A	B	C	D	E	H
1	# IDENTIFIER	ATTRIBUTE A	VALUE A	ATTRIBUTE B	VALUE B	
2	1111112111	PhoneNumber	0799998998	PublicID	+4679998999	
3	1111122111	PhoneNumber	079988999	PublicID	+46799988999	
4	1111129111	PhoneNumber	1111129111	PublicID	+46799981999	
5	1111132111	PhoneNumber	0799	PublicID	+46799978999	
6	1111134111	PhoneNumber	0779976999	PublicID	+46799976999	
7	1111211097	PhoneNumber	0799900033	PublicID	+46799900013	
8						

OverviewSummaryUnique in AUnique in BMismatch

Figure 9 Result File Example

5.2 Read a Definition

The **Definition** shows the details of the executed analysis order.

5.2.1 Rule Based Analysis Definition

Figure 10 shows an example of a rule-based-analysis order.

Definition

Details 1		Data Source A 2		Data Source B	
Name	DailyDemo	Type	Dump source	Type	Dump source
Type	Rule based analysis order	Data source	dsA.csv	Data source	dsB.csv
Occurrence	Daily 07:58:17				

Rule specification 3

	Attribute in Data Source A	Attribute in Data Source B	Rule	Identifier
>	PrivateID	PrivateID	Equal	true
▼	PhoneNumber	PublicID	Equal ending	false

Rule details

Rule	Equal ending			
Parameter	9			
>	PrivateID	SIPURI	B contains A	false

Figure 10 Rule Based Analysis Definition

In Figure 10:

- 1 Here shows the details of the analysis order. For more information, see Section 4.1 on page 11.



- 2 Here shows the data sources that the analysis order is executed on. For more information, see Section 4.2 on page 12.
- 3 Here shows the rule specification applied to the analysis order. For more information, see Section 4.4 on page 16.

5.2.2 Pattern Based Analysis Definition

The **Definition** of a pattern-based-analysis order is similar to a rule-based one, except, instead of a rule specification, the detected data relation pattern is shown in the report, as shown in Figure 11.

Pattern			
Attribute in Data Source A	Attribute in Data Source B	Distinctness Level	Identifier
Surname	Mail	99.99%	false
FamilyName	Mail	99.99%	false
PhoneNumber	PublicID	99.99%	false
PrivateID	PrivateID	100%	true
PrivateID	SIPURI	100%	false
Enabled	PrivateID	99.99%	false
Enabled	SIPURI	99.99%	false

Figure 11 Detected Patterns in Definition

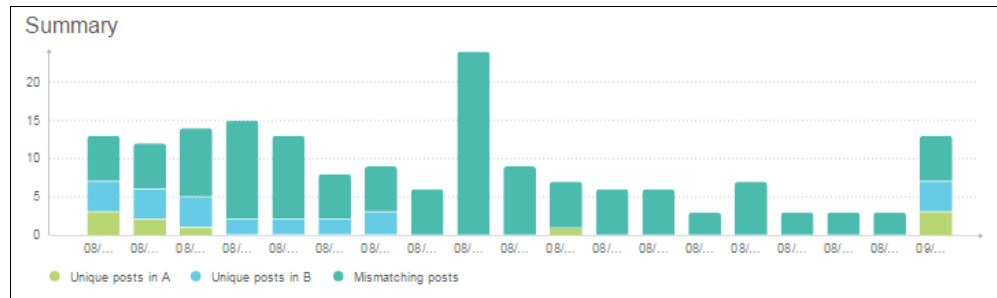
The **Distinctness Level** value indicates how clear the data relation pattern is between the attributes in **Data source A** and **B**.

Note: A recurrent pattern based order adopts the pattern changes over time. The longer a pattern has been stabilized, the longer it takes to change the pattern. Thus, it can require many analysis before a changed pattern takes effect.

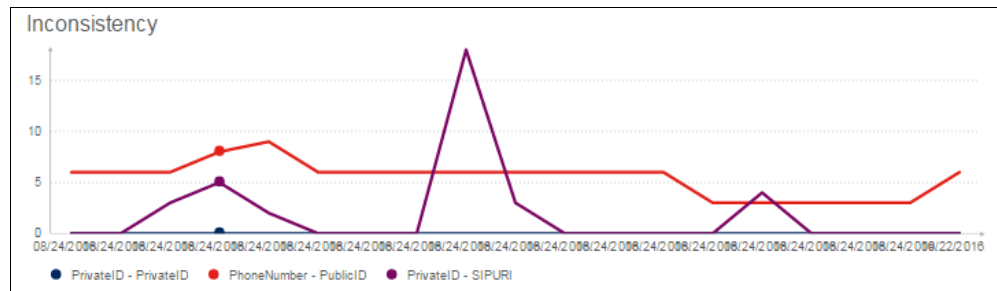
5.3 Read a Statistics

The **Statistics** report is available only for recurrent analysis orders. This report contains three parts to show the history data inconsistency status:

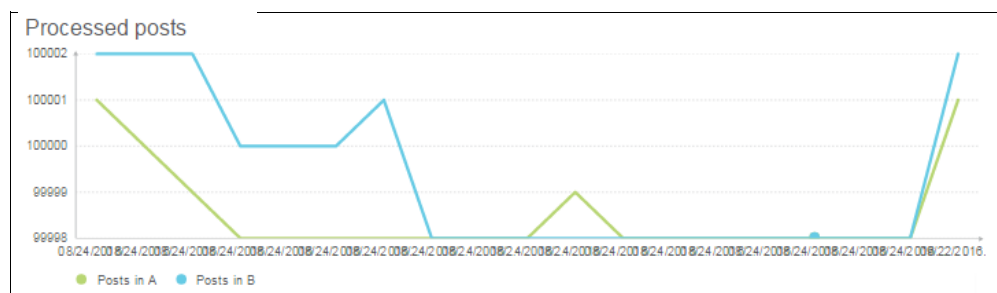
- **Summary** – Displays numbers of unique and mismatching posts in data source A and B over time.



- Each column represents one analysis execution.
- Each color represents one kind of inconsistency types: unique posts in A and B, and mismatching posts.
- **Inconsistency** – Displays the number of mismatching posts per rule.



- The rules can be either defined in a rule specification or found in the pattern-analysis.
- Each color represents one rule or a found data relation pattern.
- **Processed posts** – Displays how many posts in data source A and B.



Tips for reading a **Statistics** report:

- Hover the mouse over the graphs to show details of each execution.
- Use the timestamp to find a particular execution, and read its **Report** for further details.



6 Appendix

6.1 Comparison Rules

This section describes the comparison rules that can be used in the Consistency Checker.

6.1.1 Equal

Equal rule checks if two attributes are identical. This rule is a case-sensitive rule.

Note: When using this rule in online analysis, the attributes must be of the same type. For example, a string attribute compared against another string attribute is valid to use, while a boolean attribute compared against an integer attribute is not valid to use.

6.1.2 Equal ignore case

Equal ignore case rule is same as the **Equal** rule, except it is a case-insensitive rule.

6.1.3 Not Equal

Not Equal rule checks if two attributes are not identical. This rule is a case-sensitive rule.

Note: When using this rule in online analysis, the attributes must be of the same type. For example, a string attribute compared against another string attribute is valid to use, while a boolean attribute compared against an integer attribute is not valid to use.

6.1.4 Equal Ending

Equal ending rule checks if the last part of two compared attributes are identical characters. This rule request an additional argument to specify how many digits or chatterers should be identical.

For example, the `MSIDSN` in data source A is stored in format with country code and regional code as `0046315234562`, and in data source B in format `5234562`.

The rule argument is set to `7`, and the Consistency Checker compares the attributes from the end, counting 7 digits or characters. The rule example described here excludes the country code and regional code from the `MSIDSN` attribute from data source A.



6.1.5 A contains B

A contains B rule checks if the attribute value in data source A contains the attribute value in data source B.

For example, if the attribute in data source A is `Hello`, and the attribute in data source B is `ell`, the rule is fulfilled.

6.1.6 B contains A

B contains A rule checks if the attribute value in data source B contains the attribute value in data source A.

For example, if the attribute in data source A is `ell`, and the attribute in data source B is `Hello`, the rule is fulfilled.

6.1.7 Conditional Mapping

Conditional Mapping rule is used when comparing two attributes of different information type. For example, `STATE=CONNECTED` is compatible to `Subscriber_status=0`. This kind of rule requires a preconfigured mapping file describing the attribute pair's value mapping.

The following example shows a mapping file for attributes `STATE` and `Subscriber_state`. The mapping file should be a Java™ properties file, for more information see Reference [5].

```
# map STATE (left column) and Subscriber_status (right column) as follows
CONNECTED=1
NOT CONNECTED=0
```

Note: Spaces have to be escaped.

6.1.8 Match

The **Match** rule requires additional rule argument according to the syntax:

```
<matcher A>=<matcher B>
```

Supported matchers are:

EXIST	The attribute must have a value
!EXIST	The attribute cannot have a value
MATCH(<regEx>)	The attribute must match the provided regular expression (Java style).



The following example shows that to fulfill the rule, the attribute value from data source A must contain at least one character and the attribute value from data source B starts with ABC.

```
EXIST=MATCH (^ABC.*$)
```

6.2 Rule Conditions

This section describes the rule conditions that can be used in the Consistency Checker.

Rule conditions are entered in free-text form in the conditions field, and are not limited to the chosen attributes, and can use other attributes outside the scope of the rules. For example, a condition rule on the attributes ACC and IMSI can have conditional rules on attributes AMSISDN, state, and so on.

Syntax

The syntax of a rule condition is described as below:

<Data Source>.<Attribute> = <Conditional rule>, where:

- *<Data Source>* – Either A or B
- *<Attribute>* – A valid attribute from the chosen *<Data Source>*
- *<Conditional rule>* – Described in the following subsections.

6.2.1 Exist

The **EXIST** condition checks if an attribute exists, that is the attribute is non-empty.

```
<Data Source>.<Attribute> = EXIST
```

6.2.2 Not Exist

The **!EXIST** condition checks if an attribute does not exist, that is the attribute is empty.

```
<Data Source>.<Attribute> = !EXIST
```

6.2.3 Match

The match condition checks if an attribute matches a regular expression.

```
<Data Source>.<Attribute> = /<Regular expression>/
```




Note: Ensure that there are two “/”s, one before and one after the regular expression.

6.2.4 Logical Operators

The logical operators `AND` and `OR` are supported when defining rules.

Example 1

```
A.IMSI = EXIST AND B.MSISDN = /123*/
```

Example 2

```
A.IMSI = EXIST OR B.MSISDN = /123*/
```

Both above examples checks two conditions:

- If the `IMSI` in the current data post from data source A has a value.
- If the `MSISDN` in the current data post from data source B starts with 123.

Example 1 is successful when both two conditions are met, while Example 2 is successful when at least one condition is met.

When both operators are used in a rule, the `AND` has precedence over the `OR`.

Example 3

The value of the following expression is `false`:

```
false AND true OR false
```

The value of the following expression is `true`:

```
true OR false AND false
```




Reference List

- [1] *Installation Instruction for Consistency Checker on Glassfish Server Open Source Edition*, 5/1531-CSH 109 628 Uen
- [2] *Function Specification Consistency Checker*, 21/155 17-CSH 109 628 Uen
- [3] *System Administrators Guide for Consistency Checker*, 5/1543-CSH 109 628 Uen
- [4] *Programmers Guide for Consistency Checker*, 25/1553-CSH 109 628 Uen

Online References

- [5] *.properties*, <http://en.wikipedia.org/wiki/.properties>