CA Business Service Insight

Business Intelligence Interface Guide

8.2

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Introduction

CA Business Service Insight helps you manage and create service level agreements and reports. The material presented here comprises a reference to all user aspects of the CA Business Service Insight application.

Chapter 2: Business Intelligence Interface

The following sections describe the SQL interfaces of the Business Intelligence Interface. In addition, administration tasks such as Oracle streaming are described.

This section contains the following topics:

SQL Interfaces

The following sections describe the available SQL reporting interfaces and their supported views.

Service Level Interface

The Service Level Interface is the main reporting interface. They provide access to the calculated service level data.

The interface is made up of the following fourteen views:

- Seven basic views containing the most commonly used fields:
  - **v_psl_1_all** - Basic view of tracking period calculations.
  - **v_psl_0_hour** - Basic view of hourly calculations.
  - **v_psl_0_day** - Basic view of daily calculations.
  - **v_psl_0_week** - Basic view of weekly calculations.
  - **v_psl_0_month** - Basic view of monthly calculations.
  - **v_psl_0_quarter** - Basic view of quarterly calculations.
  - **v_psl_0_year** - Basic view of yearly calculations.
- Seven views containing additional fields for more complex usage:
  - **v_psl_all_extended** - Enhanced view of tracking period calculations.
  - **v_psl_hour_extended** - Enhanced view of hourly calculations.
  - **v_psl_day_extended** - Enhanced view of daily calculations.
  - **v_psl_week_extended** - Enhanced view of weekly calculations.
  - **v_psl_month_extended** - Enhanced view of monthly calculations.
  - **v_psl_quarter_extended** - Enhanced view of quarterly calculations.
  - **v_psl_year_extended** - Enhanced view of yearly calculations.

The views expose only records in the PSL tables that are relevant for reporting, meaning only records that meet the following criteria:

- Contain calculations of complete periods or incomplete periods, but not records that contain predictions ("best/worse").
- Contain calculations of reportable metrics.

V_PSL

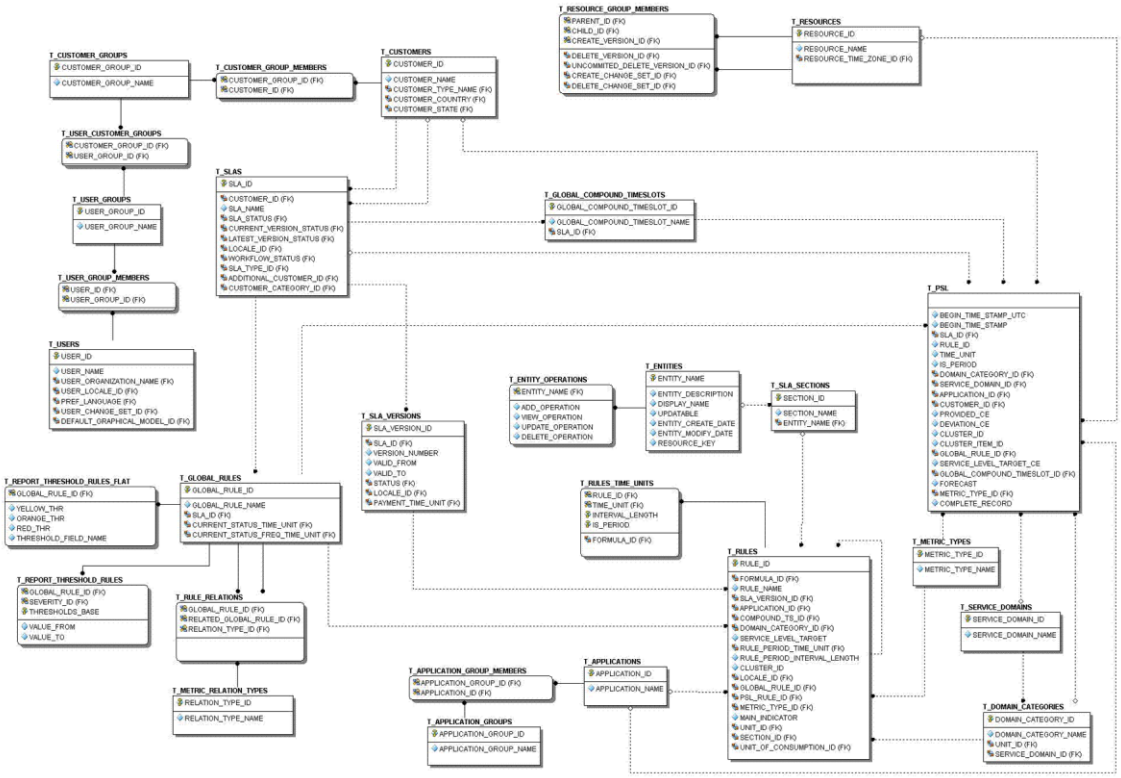
The following table describes the fields in V_PSL. This is the same for all seven views.

Name	Type	Content
BEGIN_TIME_STAMP_ UTC	date	Time stamp of the record, shifted to UTC.
BEGIN_TIME_STAMP	date	Time stamp of the record.
SLA_ID	number	Foreign Key to T_SLAS.

Name	Type	Content
GLOBAL_RULE_ID	number	Foreign Key to T_GLOBAL_RULES.
RULE_ID	number	Foreign Key to T_RULES.
PSL_RULE_ID	number	Foreign Key to T_RULES. It points to an identical canceled rule actually used to perform the calculation.
FORMULA_ID	number	Foreign Key to T_FORMULAS It points to the formula used by the rule that performs the calculations for this PSL entry.
TIME_UNIT	varchar2(30)	One of the following strings: HOUR, DAY, WEEK, MONTH, QUARTER or YEAR
IS_PERIOD	number	Refers to the sending Metric: ■ 0 - An operational calculation. ■ 1 - Business calculation according to tracking period defined in the contract.
DOMAIN_CATEGORY_ID	number	Foreign Key to T_DOMAIN_CATEGORIES.
SERVICE_DOMAIN_ID	number	Foreign Key to T_SERVICE_DOMAINS.
APPLICATION_ID	number	Foreign Key to T_APPLICATIONS.
CUSTOMER_ID	number	Foreign Key to T_CUSTOMERS.
PROVIDED_CE	number(15,5)	The Service Level value.
DEVIATION_CE	number(15,5)	Calculated deviation of the Service Level from the Target.
CLUSTER_ID	number	ID of Cluster Base Resource Group. 0 if metric is not clustered. Foreign Key to T_RESOURCES.
CLUSTER_ITEM_ID	number	ID of Cluster Resource. 0 if metric is not clustered. Foreign Key to T_RESOURCES.
SERVICE_LEVEL_ TARGET_CE	number(15,5)	The Target.
FORECAST	number(15,5)	Calculated Forecast.
METRIC_TYPE_ID	number(10)	Foreign Key to T_SLAS.

Name	Type	Content
COMPLETE_RECORD	number(1)	One of the following: ■ 0 - An ongoing calculation in the middle of the period. ■ 1 - Calculation of a period that was completed.

The following chart shows the tables and fields used by V_PSL, and the relationships between the tables.



V_PSL_EXTENDED

The following table describes the fields in V_PSL_EXTENDED. This is the same for all seven views.

Name	Type	Content
BEGIN_TIME_STAMP_ UTC	date	Time stamp of the record, shifted to UTC.
BEGIN_TIME_STAMP	date	Time stamp of the record.

Name	Type	Content
END_TIME_STAMP_UTC	date	End time stamp of the record, shifted to UTC.
END_TIME_STAMP	date	End time stamp of the record.
SLA_ID	number	Foreign Key to T_SLAS.
GLOBAL_RULE_ID	number	Foreign Key to T_GLOBAL_RULES.
RULE_ID	number	Foreign Key to T_RULES.
PSL_RULE_ID	number	Foreign Key to T_RULES. It points to an identical cancelled rule actually used to perform the calculation.
FORMULA_ID	number	Foreign Key to T_FORMULAS It points to the formula used by the rule that performs the calculations for this PSL entry.
TIME_UNIT	varchar2(30)	One of the following strings: HOUR, DAY, WEEK, MONTH, QUARTER or YEAR
INTERVAL_LENGTH	number	Number of TIME_UNITs in the calculated period.
IS_PERIOD	number	Refers to the sending Metric: ■ 0 - An operational calculation. ■ 1 - Business calculation according to tracking period defined in the contract.
DOMAIN_CATEGORY_ID	number	Foreign Key to T_DOMAIN_CATEGORIES.
SERVICE_DOMAIN_ID	number	Foreign Key to T_SERVICE_DOMAINS.
APPLICATION_ID	number	Foreign Key to T_APPLICATIONS.
CUSTOMER_ID	number	Foreign Key to T_CUSTOMERS.
EXCEPTION	number(1)	One of the following: ■ 0 - The calculated period did not include any Exceptions. ■ 1 - The calculated period included an Exception. For 1 there can be multiple Exceptions.

Name	Type	Content
CORRECTION	number(1)	One of the following: ■ 0 - The calculated period did not include any Corrections. ■ 1 - The calculated period included a Correction. For 1 there can be multiple Corrections.
PROVIDED	number(15,5)	Service Level value, calculated without Corrections or Exceptions.
PROVIDED_C	number(15,5)	Service Level value, calculated with Corrections and without Exceptions.
PROVIDED_E	number(15,5)	Service Level value, calculated without Corrections and with Exceptions.
PROVIDED_CE	number(15,5)	Service Level value, calculated with Corrections and Exceptions.
DEVIATION	number(15,5)	Calculated deviation of the Service Level from the Target, calculated without Corrections or Exceptions.
DEVIATION_C	number(15,5)	Calculated deviation of the Service Level from the Target, calculated with Corrections and without Exceptions.
DEVIATION_E	number(15,5)	Calculated deviation of the Service Level from the Target, calculated without Corrections and with Exceptions.
DEVIATION_CE	number(15,5)	Calculated deviation of the Service Level from the Target, calculated with Corrections and Exceptions.
TIME_UNIT_ORDER	number	One of the following: ■ 3 - HOUR ■ 4 - DAY ■ 5 - WEEK ■ 6 - MONTH ■ 7 - QUARTER ■ 8 - YEAR

Name	Type	Content
CLUSTER_ID	number	ID of Cluster Base Resource Group. 0 if metric is not clustered. Foreign Key to T_RESOURCES.
CLUSTER_ITEM_ID	number	ID of Cluster Resource. 0 if metric is not clustered. Foreign Key to T_RESOURCES.
GLOBAL_COMPOUND_ TIMESLOT_ID	number	Foreign Key to T_GLOBAL_COMPOUND_TIMESLOTS.
PSL_CREATE_DATE	date	Creation date of the record.
PSL_MODIFY_DATE	date	Modification date of the record.
SERVICE_LEVEL_ TARGET	number(15,5)	The Target, in the calculation without Corrections or Exceptions.
SERVICE_LEVEL_ TARGET_C	number(15,5)	The Target, in the calculation with Corrections and without Exceptions.
SERVICE_LEVEL_ TARGET_E	number(15,5)	The Target, in the calculation without Corrections and with Exceptions.
SERVICE_LEVEL_ TARGET_CE	number(15,5)	The Target, in the calculation with Corrections and Exceptions.
FORECAST	number(15,5)	Calculated Forecast.
METRIC_TYPE_ID	number(10)	Foreign Key to T_METRIC_TYPES.
COMPLETE_RECORD	number(1)	One of the following: ■ 0 - An ongoing calculation in the middle of the period. ■ 1 - Calculation of a period that was completed.

Raw Data and Intermediate Data Interfaces

The Raw Data and Intermediate Data interfaces supply access to the input events for calculations:

1. The Raw Data Interface exposes the events received from Adapters and:
    a. Can handle deletion of records
    b. When the Raw Data decoding process encounters events that have a corrupted value in the encoded data field, it skips the problematic record, writes an entry in the process log and continues.
    c. Raw Data decoding process also handles updates of existing events and applies them to the T_RAW_DATA_FIELDS table. This insures that no discrepancies occur as a result of correction to the T_RAW_DATA table.
2. The Intermediate Data Interface exposes events generated by calculations and:
    a. Handles deletion of records.
    b. When the Intermediate Data decoding process encounters events that have a corrupted value in the encoded data field, it skips the problematic record, writes an entry in the process log and continues.

Both the Raw Data and Intermediate Data decoding processes are able to clean up log entries from the T_RAW_DATA_FIELDS_LOG table.

In addition, both processes cleanup their respective log entries (i.e. the Raw Data decoding process cleans up raw data decoding log entries; the Intermediate Data decoding process cleans up intermediate data decoding log entries).

Log entries are considered candidates for cleanup according to:

- Number of days since they were written
- Severity (Debug, Info, Error)
- Configuration parameters. The parameters (see below) are located in the T_RAW_DATA_FIELDS_CONFIG table:
    - Debug – How long (in days) to keep entries of severity Debug. Default value: 30
    - Info – How long (in days) to keep entries of severity Info. Default value: 90
    - Error – How long (in days) to keep entries of severity Error. Default value: 365

Event data is supported in HTML format.

V_RD_<eventtype>

For each Event Type in the system, a view named V_RD_<eventtype> is created (where <eventtype> is the name of the event type for which the view is created).

- This view exposes records of the specific Event Type.
- Instead of an XML that contain the data of the Event, this view has an actual column for each Field of the Event. The column name is according to the name of the Field, as defined in the system, and the column type is a string.

These views can be useful when designing reports on a specific Event Type.

The V_RD_<eventtype> interface is able to handle raw data fields with their real types, according to the definition of the event type.

If the Event name and/or the Event Field names contain non-ASCII characters, they are replaced by '_', or removed from the name.

Names longer than 30 characters are truncated. In case the truncation returns a name that already exists, the name is changed to end with a sequence number.

The following table describes the fields in V_RD_<eventtype>.

Name	Type	Content
READER_ID	number	Foreign Key to T_READERS.
READER_NAME	varchar2(30)	Name of the Adapter.
RESOURCE_ID	number	Foreign Key to T_RESOURCES.
RESOURCE_NAME	varchar2(100)	Name of the Resource.
TIME_STAMP	date	Time stamp of the event (UTC time).
READER_TIME_STAMP	date	Time stamp of when the Adapter created the event (UTC time).
CREATED	number	If the event was added by a Correction, contains the ID of the correction.
CORRECTED	number	If the event was deleted by a Correction, contains the ID of the correction.
CREATE_DATE	date	The creation date of the record.
MODIFY_DATE	date	The modification date of the record.
ANNOTATION_USER_ ID	number	Foreign Key to T_USERS. If the event has an annotation, points to the user who created the annotation.

Name	Type	Content
ANNOTATION_USER_NAME	varchar2(200)	If the event has an annotation, contains the name of the user who created the annotation.
ANNOTATION_TEXT	varchar2(512)	If the event has an annotation, contains the annotation's text.
ANNOTATION_IS_MANUAL	number	If the event has an annotation: ■ 0 - Annotation created by Business Logic. ■ 1 - Annotation created manually by user.
ANNOTATION_MODIFY_DATE	date	If the event has an annotation, contains the modification date of the annotation.
F_NAME1	field1type	Value of field #1 in the Event.
F_NAME2	field2type	Value of field #2 in the Event.
F_NAMEX	fieldxtype	Value of field #x in the Event.

V_ID_<eventtype>

For each Event Type in the system, a view named V_ID_<eventtype> is created (where <eventtype> is the name of the event type for which the view is created).

- This view exposes records of the specific Event Type.
- Instead of an XML that contain the data of the Event, this view has an actual column for each Field of the Event. The column name is according to the name of the Field, as defined in the system and the column type is a string.

These views can be useful when designing reports on a specific Event Type.

The V_ID_<eventtype> interface is able to handle raw data fields with their real types, according to the definition of the event type.

In case the Event name and/or the Event Field names contain non-ASCII characters, they are replaced by '_', or removed from the name.

Names longer than 30 characters are truncated. In case the truncation returns a name that already exists, the name is changed to end with a sequence number.

The following table describes the fields in V_ID_<eventtype>.

Name	Type	Content
METRIC_ID	number(10)	Foreign Key to T_RULES.

Name	Type	Content
METRIC_GLOBAL_ID	number(10)	Foreign Key to T_GLOBAL_RULES.
METRIC_GLOBAL_NAME	varchar2(200)	Name of the Metric that sent the Event.
RESOURCE_ID	number(10)	Foreign Key to T_RESOURCES. Null if the Event is not related to a specific Resource.
RESOURCE_NAME	varchar2(100)	Name of the Resource. Null if the Event is not related to a specific Resource.
TIME_UNIT	varchar2(30)	Refers to the sending Metric: HOUR, DAY, WEEK, MONTH, QUARTER, or YEAR (for doing a join with V_PSL_EXTENDED on page).
IS_PERIOD	number(1)	Refers to the sending Metric: ■ 0 - Operational calculation. ■ 1 - Business calculation according to the tracking period defined in the contract.
COMPLETE_PERIOD	number(1)	Refers to the sending Metric: ■ 0 - Ongoing calculation in the middle of the period. ■ 1 - Calculation of a period that was completed.
INTERVAL_LENGTH	number(10)	Refers to the sending Metric: the number of TIME_UNITs in the calculated period.
TRG_TIME_STAMP	date	The time stamp of the Event (UTC time).
WITH_EXCEPTION	number(1)	■ 0 - Calculation that ignores Exceptions. ■ 1 - Calculation that takes Exceptions into account.
WITH_CORRECTION	number(1)	■ 0 - Calculation that ignores Corrections. ■ 1 - Calculation that takes Corrections into account.

Name	Type	Content
CORRECTION_FOLDED	number(1)	■ 0 - Correction took place, and the current calculation was performed with and without Corrections separately. ■ 1 - No Correction took place, and the current calculation was performed with and without Corrections together.
EXCEPTION_FOLDED	number(1)	■ 0 - Exception took place, and the current calculation was performed with and without Exceptions separately. ■ 1 - No Exception took place, and the current calculation was performed with and without Exceptions together.
CREATE_DATE	date	Creation date of the record.
MODIFY_DATE	date	Modification date of the record.
CALCULATION_MODE	varchar2(20)	Either NORMAL or FORECAST.
F_NAME1	field1type	Value of field #1 in the Event.
F_NAME2	field2type	Value of field #2 in the Event.
F_NAMEX	fieldxtype	Value of field #x in the Event.

Additional Calculated Data Interface

V_SLALOM_OUTPUTS

This view exposes all SLALOM OUTPUT records written by business logic.

The following table describes the fields in V_SLALOM_OUTPUTS.

Name	Type	Content
RECORD_KEY	varchar2(50)	Unique identifier of the record.
RECORD_TIME_STAMP	date	Time stamp of the record (UTC time).
TABLE_NAME	varchar2(30)	Name of the logical table that the record belongs to.
RULE_ID	number	Foreign Key to T_RULES.

Name	Type	Content
CLUSTER_ITEM_ID	number	Foreign Key to T_RESOURCES. If the Metric that created the record is clustered, this field contains the ID of the Cluster Resource.
IS_PERIOD	number(1)	Refers to the sending Metric: ■ 0 - Operational calculation. ■ 1 - Business calculation according to the tracking period defined in the contract.
COMPLETE_PERIOD	number(1)	Refers to the sending Metric: ■ 0 - Ongoing calculation in the middle of the period. ■ 1 - Calculation of a period that was completed.
CREATE_DATE	date	Creation date of the record.
MODIFY_DATE	date	Modification date of the record.
RESOURCE_ID	number	Foreign Key to T_RESOURCES.
RESOURCE_NAME	varchar2(100)	Name of the Resource.
VAL_1	varchar2(512)	Value of field #1 of the record.
VAL_2	varchar2(50)	Value of field #2 of the record.
VAL_X	varchar2(50)	Value of field #x of the record.

V_SO_<tablename>

For each TABLE_NAME in T_SLALOM_OUTPUTS, a view named V_SO_<tablename> is created (where <tablename> is the name of the table for which the view is created).

For the different values of the TABLE_NAME field from the T_SLALOM _OUTPUTS table, in the table T_SO_FIELD_ NAMES definition for the data structure in the VAL_X fields is given.

Similar to V_RD_<eventtype>, every view defined for the specific TABLE_NAME value exposes different fields. The views have all the fields that are in the V_SLALOM_OUTPUTS view, except that the VAL_X fields are replaced with custom named fields that depend on the field definitions for the table name in the T_SO_TABLE_NAMES table.

In case the Table name and/or the Field names contain non-ASCII characters, they are replaced by '_', or removed from the name, the same as for V_RD_<eventtype>.

Names longer than 30 characters are truncated. In case the truncation returns a name that already exists, the name is changed to end with a sequence number, the same as for V_RD_<eventtype>.

It is possible that in the T_SLALOM_OUTPUT table there exist TABLE_NAMEs for which there is no information for the data structure in the T_SO_FIELD_NAMES table. In such a case the View is still created, but instead of custom field names, the field names follow the VAL_X naming convention.

V_BREACH_ROOT_CAUSE

This view exposes all Breach Root Cause records written by business logic or manually by users.

The following table describes the fields in V_BREACH_ROOT_CAUSE.

Name	Type	Content
RECORD_KEY	varchar2(50)	Unique identifier of the record.
RECORD_TIME_STAMP	date	Time stamp of the record (UTC time).
RULE_ID	number	Foreign Key to T_RULES.
CLUSTER_ITEM_ID	number	Foreign Key to T_RESOURCES. If the Metric that created the record is clustered, this field contains the ID of the Cluster Resource.

Name	Type	Content
IS_PERIOD	number(1)	Refers to the sending Metric: ■ 0 - Operational calculation. ■ 1 - Business calculation according to the tracking period defined in the contract.
COMPLETE_PERIOD	number(1)	Refers to the sending Metric: ■ 0 - Ongoing calculation in the middle of the period. ■ 1 - Calculation of a period that was completed.
CREATE_DATE	date	Creation date of the record.
MODIFY_DATE	date	Modification date of the record.
RESOURCE_ID	number	Foreign Key to T_RESOURCES.
RESOURCE_NAME	varchar2(100)	Name of the Resource.
DESCRIPTION	varchar2(512)	Description written by the logic/user.

V_FORMULA_TO_RAW_DATA

This view enables the user to retrieve raw data events while filtering those events according to the formulas that used them.

The view exposes the field FORMULA_ID which is a foreign key to T_FORMULAS and in addition the 112 fields that make up the table T_RAW_DATA_FIELDS:

Name	Type	Content
FORMULA_ID	number	Foreign Key to T_FORMULAS It points to a formula that uses this event.
RAW_DATA_ID	number	The ID of the event in T_RAW_DATA
READER_ID	number	The ID of the reader (adapter) which read this event
EVENT_TYPE_ID	number	The ID of the type of this event
RESOURCE_ID	number	The ID of the resource of this event
TIME_STAMP	date	The time stamp of this event
READER_TIME_ STAMP	date	The time stamp of the reader of the event

Name	Type	Content
CREATED_BY	number	0 if the event is a regular event read by an adapter or of this event is a result of a correction this would be the correction ID
CORRECTED_BY	number	0 if this event is relevant and there is no correction overriding it any other number would be a correction id which overrides this event
CREATE_DATE	date	The timestamp at which this event was entered to the system
MODIFY_DATE	date	The timestamp of the last change to this event
DATA_SOURCE_ID	varchar2(60)	The name of the input file from which the event is retrieved and that the adapter reads.
HASH_DATA_KEY	varchar2(16)	A value used to distinguish between events for sake of event singularity
DATA_HTML	CLOB	Aggregation of all values of all the dynamic fields of the event in HTML format including field names.
HTML_DATA_STR	varchar2(4000)	The first 4000 characters of the HTML data. **Note**: This field content might not be complete if the event data is longer than 4000 characters.
FIELD_001	varchar2(4000)	The 1st value of the first dynamic field of this event
FIELD_002	varchar2(4000)	The 2nd value of the first dynamic field of this event
FIELD_099	varchar2(4000)	The 99th value of the first dynamic field of this event
FIELD_100	varchar2(4000)	The 100th value of the first dynamic field of this event

Each event has an event type and each event type has a predefined structure which dictates how many (if at all) dynamic fields it has, their names, order and type. So one event of a certain type may have three dynamic fields and another may have 10 fields.

Each event in this view has 100 such fields named field_000 – field_100 but only the ones relevant to the event in question according to its event type have meaningful values in them, the rest hold the a null value.

Furthermore those fields may hold values of one of the following four types – integer, float, string & date. However as the same set of 100 fields should be able to hold event fields of events of different event types and as such with different data types, they are all treated as strings and represented in the views as varchar2 columns.

The usual way one would use this view would be to retrieve all events consumed by a certain metric, probably for a certain period of time.

So if we would want to retrieve all the raw data events that were taken into account in the calculation of a certain service level or, in other words, we would wanted to make the correlation between a record in one of the v_psl views and a record in the t_raw_data_fields table we would do the following:

1. Retrieve the formula_id from the relevant v_psl view.
2. Use this formula_id to filter the relevant records from v_formula_to_raw_data.

   We may also want to further filter the events according to their timestamp so as to achieve the appropriate correlation between a set of PSL records at a certain timeframe and the set of events that have been consumed by the engine during the calculation of those PSL entries.

V_FORMULA_TO_INTERMEDIATE_DATA

This view enables the user to retrieve intermediate data events while filtering those events according to the formulas that used them.

The view exposes the field FORMULA_ID which is a foreign key to T_FORMULAS and in addition the 112 fields that make up the table T_INTERMEDIATE_DATA_FIELDS:

Name	Type	Content
FORMULA_ID	number	Foreign Key to T_FORMULAS It points to a formula that uses this event.
METRIC_EVENT_ID	number	The ID of the event in T_INTERMEDIATE_DATA.
METRIC_ID	number	Foreign Key to T_RULES.
METRIC_GLOBAL_ID	number	Foreign Key to T_GLOBAL_RULES.
EVENT_TYPE_ID	number	Foreign Key to T_EVENT_TYPES.

Name	Type	Content
RESOURCE_ID	number	Foreign Key to T_RESOURCES. Null if the Event is not related to a specific Resource.
TIME_UNIT	varchar2(30)	Refers to the sending Metric: HOUR/DAY/WEEK/MONTH/QUARTER/ YEAR (for doing a join with V_PSL_EXTENDED).
IS_PERIOD	number	Refers to the sending Metric: ■ 0 - Operational calculation. ■ 1 - Business calculation according to the tracking period defined in the contract.
COMPLETE_PERIOD	number	Refers to the sending Metric: ■ 0 - Ongoing calculation in the middle of a period. ■ 1 - Calculation of a period that was completed.
INTERVAL_LENGTH	number	Refers to the sending Metric: the number of TIME_UNITs in the calculated period.
SRC_TIME_STAMP	date	The time stamp of the event was written.
TRG_TIME_STAMP	date	Time stamp of the Event (UTC time).
WITH_EXCEPTION	number	■ 0 - Calculation that ignores Exceptions. ■ 1 - Calculation that takes Exceptions into account.
WITH_CORRECTION	number	■ 0 - Calculation that ignores Corrections. ■ 1 - Calculation that takes Corrections into account.
CORRECTION_FOLDED	number	■ 0 - Correction took place, and the current calculation was performed with and without Corrections separately. ■ 1 - No Correction took place, and the current calculation was performed with and without Corrections together.

Name	Type	Content
EXCEPTION_FOLDED	number	■ 0 - Exception took place, and the current calculation was performed with and without Exceptions separately. ■ 1 - No Exception took place, and the current calculation was performed with and without Exceptions together.
CREATE_DATE	date	Creation date of the record.
MODIFY_DATE	date	Modification date of the record.
CALCULATION_MODE	varchar2(20)	Either NORMAL or FORECAST.
DATA_HTML	CLOB	Aggregation of all values of all the dynamic fields of the event in HTML format including field names.
HTML_DATA_STR	varchar2(4000)	The first 4000 characters of the HTML data. **Note**: This field content might not be complete if the event data is longer than 4000 characters.
FIELD_001	varchar2(4000)	The 1st value of the first dynamic field of this event
FIELD_002	varchar2(4000)	The 2nd value of the first dynamic field of this event
FIELD_099	varchar2(4000)	The 99th value of the first dynamic field of this event
FIELD_100	varchar2(4000)	The 100th value of the first dynamic field of this event

Each event has an event type and each event type has a predefined structure which dictates how many (if at all) dynamic fields it has, their names, order and type. So one event of a certain type may have three dynamic fields and another may have 10 fields.

V_FORMULA_TO_DATA

This view holds data that is a combination of the V_FORMULA_TO_RAW_DATA and V_FORMULA_TO_INTERMEDIATE_DATA views, where raw data events have the value RD under the column DATA_TYPE and intermediate data events have the value ID in that field.

Note: Columns that are specific for Raw Data events have a NULL value in the intermediate data specific columns and vice a versa.

Name	Type	Content	Source of Column
FORMULA_ID	number	Foreign Key to T_FORMULAS It points to a formula that uses this event.	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
EVENT_ID	number	The ID of the event in T_INTERMEDIATE_ DATA or T_RAW_ DATA.	V_FORMULA_TO_ RAW_DATA (RAW_DATA_ID) or V_FORMULA_TO_ INTERMEDIATE_DATA (METRIC_EVENT_ID)
DATA_TYPE	varchar2(2)	Indication of the whether the data event is Raw Data (RD) or Intermediate Data (ID)	NA
EVENT_TYPE_ID	number	Foreign Key to T_EVENT_TYPES.	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
RESOURCE_ID	number	Foreign Key to T_RESOURCES. Null if the Event is not related to a specific Resource.	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
TIME_STAMP	date	Time stamp of the Event (UTC time).	V_FORMULA_TO_ RAW_DATA (TIME_STAMP) or V_FORMULA_TO_ INTERMEDIATE_DATA (TRG_TIME_STAMP)

Name	Type	Content	Source of Column
SOURCE_TIME_ STAMP	date	The time stamp of the event was written.	V_FORMULA_TO_ RAW_DATA (READER_TIME_STAMP) or V_FORMULA_TO_ INTERMEDIATE_DATA (SRC_TIME_STAMP)
CREATE_DATE	date	Creation date of the record.	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
MODIFY_DATE	date		This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
DATA_HTML	CLOB	Aggregation of all values of all the dynamic fields of the event in HTML format including field names.	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
HTML_DATA_ STR	varchar2(4000)	The first 4000 characters of the HTML data. Note: This field content might not be complete if the event data is longer than 4000 characters.	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
READER_ID	number	Foreign Key to T_READERS.	V_FORMULA_TO_ RAW_DATA
CREATED_BY	number	If the event was added by a Correction, contains the ID of the Correction.	V_FORMULA_TO_ RAW_DATA

Name	Type	Content	Source of Column
CORRECTED_BY	number	If the event was deleted by a Correction, contains the ID of the Correction.	V_FORMULA_TO_ RAW_DATA
DATA_SOURCE_ ID	varchar2(60)	The name of the input file from which the event is retrieved and that the adapter reads.	V_FORMULA_TO_ RAW_DATA
HASH_DATA_ KEY	varchar2(16)	A value used to distinguish between events for sake of event singularity.	V_FORMULA_TO_ RAW_DATA
METRIC_ID	number	Foreign Key to T_RULES	V_FORMULA_TO_ INTERMEDIATE_DATA
METRIC_ GLOBAL_ID	number	Foreign Key to T_GLOBAL_RULES	V_FORMULA_TO_ INTERMEDIATE_DATA
TIME_UNIT	varchar2(30)	Refers to the sending Metric: HOUR/DAY/WEEK/MONTH/QUARTER/YEAR (for doing a join with V_PSL_EXTENDED).	V_FORMULA_TO_ INTERMEDIATE_DATA
IS_PERIOD	number	Refers to the sending Metric: ■ 0 - Operational calculation. ■ 1 - Business calculation according to the tracking period defined in the contract.	V_FORMULA_TO_ INTERMEDIATE_DATA

Name	Type	Content	Source of Column
COMPLETE_ PERIOD	number	Refers to the sending Metric: ■ 0 - Ongoing calculation in the middle of a period. ■ 1 - Calculation of a period that was completed.	V_FORMULA_TO_ INTERMEDIATE_DATA
INTERVAL_ LENGTH	number	Refers to the sending Metric: the number of TIME_UNITs in the calculated period.	V_FORMULA_TO_ INTERMEDIATE_DATA
WITH_ EXCEPTION	number	■ 0 - Calculation that ignores Exceptions. ■ 1 - Calculation that takes Exceptions into account.	V_FORMULA_TO_ INTERMEDIATE_DATA
WITH_ CORRECTION	number	■ 0 - Calculation that ignores Corrections. ■ 1 - Calculation that takes Corrections into account.	V_FORMULA_TO_ INTERMEDIATE_DATA

Name	Type	Content	Source of Column
CORRECTION_ FOLDED	number	■ 0 - Correction took place, and the current calculation was performed with and without Corrections separately. ■ 1 - No Correction took place, and the current calculation was performed with and without Corrections together.	V_FORMULA_TO_ INTERMEDIATE_DATA
EXCEPTION_ FOLDED	number	■ 0 - Exception took place, and the current calculation was performed with and without Exceptions separately. ■ 1 - No Exception took place, and the current calculation was performed with and without Exceptions together.	V_FORMULA_TO_ INTERMEDIATE_DATA
CALCULATION_ MODE	varchar2(20)	Either NORMAL or FORECAST.	V_FORMULA_TO_ INTERMEDIATE_DATA
FIELD_001	varchar2(4000)	The 1st value of the first dynamic field of this event	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA

Name	Type	Content	Source of Column
FIELD_002	varchar2(4000)	The 2nd value of the first dynamic field of this event	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
FIELD_099	varchar2(4000)	The 99th value of the first dynamic field of this event	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA
FIELD_100	varchar2(4000)	The 100th value of the first dynamic field of this event	This field exists in both V_FORMULA_TO_ RAW_DATA and V_FORMULA_TO_ INTERMEDIATE_DATA

V_RULE_CLUSTER

This view enables the user to retrieve the ID and name of the resource on the basis of which a rule is clustered.

Name	Type	Content
rule_id	number	The ID of the rule
rule_name	varchar2(200)	The name of the rule
cluster_id	number	The ID of the resource on top of which this rule is clustered
cluster_name	varchar2(100)	The name of the resource on top of which this rule is clustered

V_RULE_CLUSTER_ITEMS

This view enables the user to retrieve the list of cluster members (items) of a rule.

Name	Type	Content
rule_id	number	The ID of the rule who's cluster members we are interested in
rule_name	varchar2(200)	The name of the rule who's cluster members we are interested in
cluster_item_id	number	The ID of a member resource of the cluster of the rule of interest

Name	Type	Content
cluster_item_name	varchar2(100)	The name of a member resource of the cluster of the rule of interest
cluster_item_is_group	number	■ 1 if this member is a resource group itself (relevant for recursive clusters including non leaf nodes) ■ 0 if this member is an actual resource

Utility Functions

The Utility Functions interface enables users to set the correct decimal separator and date format in an Objective Statement text. In addition, users can use this interface to format Table Parameters as tables.

FNC_PARSE_OBJECTIVE_STATEMENT

Description

Used for setting the correct decimal separator and date format in an Objective Statement text.

Syntax

```
FNC_PARSE_OBJECTIVE_STATEMENT (TEXT, SEPARATOR, DATE_FORMAT)
```

Parameters

Name	Type	Description
Text	objective_statement_text	The objective statement text.
Separator	char	The decimal separator to use.
Date_Format	varchar2	The date format to use.

Return Value

Returns the Objective Statement, with all numbers formatted with the selected decimal separator, and with all dates formatted with the selected date format.

Remarks

The default decimal separator is a period (**.**).

Example

```
FNC_PARSE_OBJECTIVE_STATEMENT ('No less than #99.97% per each Month', ',',
'MM/DD/YYYY')
```

returns

```
'No less than 99,97% per each Month'
```

FNC_PARSE_TABLE_PARAMETER

Description

Used for formatting a Table Parameter as a table.

Syntax

```
FNC_PARSE_TABLE_PARAMETER (TEXT)
```

Parameters

Name	Type	Description
TEXT	table_value	The text of the internal XML representation of a Table Parameter.

Return Value

Returns a table like the text containing the data stored in the Table Parameter.

Remarks

None.

Example

```
FNC_PARSE_TABLE_PARAMETER (TEXT)
```

where TEXT contains:

```
<Table>
  <TableStructure>
    <Col Name="c1string" Index="Y" Mandatory="N" Type="Text" DefaultValue="" />
    <Col Name="c2number" Index="N" Mandatory="N" Type="Number" DefaultValue="" />
    <Col Name="c3date" Index="N" Mandatory="N" Type="Date" DefaultValue="" />
  </TableStructure>
  <TableElements>
    <Elem>
      <c1string>str1</c1string>
      <c2number>1</c2number>
      <c3date>05/02/2008 00:00:00</c3date>
    </Elem>
    <Elem>
      <c1string>str2</c1string>
      <c2number>2</c2number>
      <c3date>28/02/2008 00:00:00</c3date>
    </Elem>
  </TableElements>
</Table>
```

returns the following table:

c1string	c2number	c3date
str1	1	05/02/2008 00:00:00
str2	2	28/02/2008 00:00:00

Exposed Tables

The BI Interface database has two database users.

The first database user, FULL_USER is used to hold all the replicated tables and all the interface views and tables that are going to be created in the database schema of this user.

Not all of this data is needed when creating and designing reports, which is why a second database user, REGULAR_USER, is created. This second user has view access rights to all the tables and views except for: Full T_PSL interface, T_PSL, T_RAW_DATA, T_SLALOM_OUTPUTS, T_INTERMEDIATE_DATA.

The list of tables that need to be replicated is defined as a configuration parameter.

Administration

Administration options enable you to:

- Use the Oracle Streams feature to propagate data, transactions and events in a data stream.
- Use a configuration tool that generates the SQL scripts necessary to install the system.
- Duplicate the database.
- Configure the Oracle Streams.

Oracle Streams

Oracle Streams, a built-in feature of the Oracle database, is a data replication and integration feature. It provides a flexible infrastructure that meets a wide variety of information sharing needs. Oracle Streams enables the propagation of data, transactions and events in a data stream either within a database, or from one database to another.

For additional information about replication technology, see the Oracle website.

Note: The Oracle Streams feature is only relevant when configuring the BI Interface for installation on two separate databases.

Initial Configuration Tool

The initial configuration tool is a part of the installation process. It enables the installation to be configurable, by generating SQL scripts needed for installing the system that take into account the configuration parameters. After all initial configuration parameters are set, this tool generates the SQL scripts that the user needs to execute in order to install the system.

The RDLProperties.xml file contains all the relevant parameters needed for the creation of the needed scripts. See Configuration for details on defining the initialization parameters in the RDLProperties.xml file.

When configuring the BI Interface for installation on two separate databases, the installation assumes that the target DB exists, but that the streams DB users and the second target DB user do not exist. The computer where the scripts will run needs to have access to both databases through sqlplus with SYS privileges for connection "as sysdba". Also, batch files for making the export and the import will be created, and these batch files should be executed before the rest of the scripts.

The configuration tool consists of the following parts:

- The scripts that are needed for creating the replication processes and interface views and functions. These scripts are general and will contain parameters for the variable sections of the scripts.
- A simple tool that parses the initial scripts and replace the parametric parts of the scripts with the configuration parameters from the parameters file.
- The batch files for executing the scripts that will be generated after parsing. The scripts will also create log files containing the log of the script execution and any problems encountered during the script execution. The batch files will return a list of problems encountered during script execution.
- A tool for creating export and import commands for the needed tables according to the properties file.
- An undo batch. Batch files that execute the scripts that remove the Oracle Streams environment; delete the created users, views, and functions; and put the databases back to the pre-install condition. There are two versions of these undo scripts, one for Oracle version 10 and later, and one for Oracle 9.

Duplicating the Database

Notes:

- This section is only relevant when configuring the BI Interface for installation on two separate databases
- This section is not executed by the user. The operations described here are performed automatically by scripts, and this is only an explanation of how they are implemented.

Because the system uses a capture process to replicate changes made to a database object from a source database to a target database, the target database must have a copy of the database object.

If the copy does not exist on the target database, the object must be instantiated on the target database before it can replicate changes. To instantiate an object means to create an object physically on a target database, based on an object on a source database. If the object to be instantiated is a table, then the objects at the source and destination databases need not be an exact match. However, if some or all of the table data is replicated between the two databases, then the data that is replicated should be consistent when the table is instantiated.

This instantiation is performed using the export/import utilities of the Oracle database. For the export of the objects, it is important that they to be consistent at a single point in time. This can be accomplished either by setting the export parameter OBJECT_CONSISTENT to 'y', or by setting the database in a read-only mode while exporting the data. When importing the needed object for the data replication, the parameter STREAMS_INSTANTIATION must be set to 'Y'.

Configuring the Streams

Notes:

- This section is only relevant when configuring the BI Interface for installation on two separate databases.
- This section is not executed by the user. The operations described here are performed automatically by scripts, and this is only an explanation of how they are implemented.

The system uses Oracle streams for the replication process. To perform data replication using Oracle Streams, the following is required:

- Create Stream Administrator users on both databases.
- Create a capture process on the source database that captures the changes the user wants to replicate to the target database.
- Create a queue that holds the captured changes on the source database.
- Create a propagation process that propagates the changes recorded in the queue.
- Create a queue on the target database that receives the changes propagated from the source database.
- Create an apply process that applies the changes from the queue.
- Create capture rules to define which tables should be replicated. These rules depend on the initial configurations for the tables selected for replication.

Chapter 3: Installation on One Database

Installation consists of the following machines:

- CA Business Service Insight database – The CA Business Service Insight database that is used for the BI Interface installation.
- BI Installation – An optional machine that is used to create and run the installation scripts from instead of from the CA Business Service Insight database.

The installation process is composed of the following steps:

1. Performing prerequisites
2. Configuration
3. Installation Process
4. Post Installation Activities

This section contains the following topics:

Prerequisites

- Verify that you have enough disk space (approximately 30% of the source database size) on the machine you are working from (CA Business Service Insight source database or the BI Installation machine) for the dump file.
- Verify that the machine you are working from (CA Business Service Insight database or the BI Installation machine) has Java Runtime Environment, version 1.5 or later, installed on it. This is required so that the batch files for installing the replication environment can be executed. Additional information on how to install and set up the Java Runtime environment can be found at the Java web site.
- Verify that you have a connection from the machine you are working from (the BI Installation machine) to the CA Business Service Insight database for sys user (e.g. sys/sys as sysdba) and check for proper connectivity to this instance using SQLPLUS (e.g. conn sys/sys@<DBNAME>).
- Stop all CA Business Service Insight services if any exist. Verify that there is no mechanism that can start them up automatically.

Configuration

Prior to installation, edit the RDLProperties.xml file, located under the BI_Interface folder, as follows:

- In the Source DB Users section edit the following keys according to your specifications:
  - SourceGlobalName: The global name of the source database.
  - Source_TNSNAME: The alias needed to connect to the source database from where the scripts are run.
  - SourceDBUser: An existing DB user on the source database where the CA Business Service Insight tables exist.
  - SourceDBPass: The password for SourceDBUser.
  - SourceDBAdminUser: The system user of the source database.
  - SourceDBAdminPass: The password for SourceDBAdminUser.
  - SourceStreamAdminUser: This user is created by the SQL scripts, therefore a user with this name should not exist. In this type of installation, keep the default value for this parameter.
  - SourceStreamAdminPass: The password for SourceStreamAdminUser. In this type of installation, keep the default value for this parameter.
- In the Target DB Users section edit the following keys according to your specifications:
  - TargetGlobalName: The same value set in SourceGlobalName.
  - Target_TNSNAME: The same value set in Source_TNSNAME.
  - TargetDBUser: The same value set in SourceDBUser.
  - TargetDBPass: The same value set in SourceDBPass.
  - TargetDBAdminUser: The same value set in SourceDBAdminUser.
  - TargetDBAdminPass: The same value set in SourceDBAdminPass.
  - TargetDBRegularUser: This user is created. This user is granted read-only access to the replicated data and the created views, excluding: full T_PSL interface, T_PSL, T_RAW_DATA, T_SLALOM_OUTPUTS, and T_INTERMEDIATE_DATA.
  - TargetDBRegularPass: The password for the TargetDBRegularUser.
  - TargetStreamAdminUser: This user will be created by the SQL scripts, therefore a user with this name should not exist. In this type of installation, keep the default value for this parameter.
  - TargetStreamAdminPass: The password for the TargetStreamAdminUser. In this type of installation, keep the default value for this parameter.
  - FilePath: Specifies the location of the database file for new table space.

- In the Tables to be replicated and interfaces to be created section edit the following keys according to your specifications:
  - Replication: Specifies whether installation is to be on two separate databases connected using streams technology (true) or on the original database (false). In this type of installation set the parameter to false.
  - ReplicatedTables_PSL: List of tables needed to be replicated for the T_PSL interface.
  - CreatePSLInterface: Specifies whether the T_PSL interface should be created. Possible values are true or false.
  - ReplicatedTables_RAW_DATA: List of tables needed to be replicated for the T_RAW_DATA interface.
  - CreateRAW_DATAInterface: Specifies whether the T_RAW_DATA interface should be created. Possible values are true or false.
  - ReplicatedTables_SLALOM_OUTPUTS: List of tables needed to be replicated for the T_SLALOM_OUTPUTS interface.
  - CreateSLALOM_OUTPUTSInterface: Specifies whether the T_SLALOM_OUTPUTS interface should be created. Possible values are true or false.
  - ReplicatedTables_Booklets: List of tables needed to be replicated for the Booklet mapping interface.
  - CreateBookletsInterface: Specifies whether the Booklet mapping interface should be created. Possible values are true or false.
  - AdditionalTables: List of any additional tables that need to be included in the replication process.
  - AdditionalTablesInterface: Specifies whether the Additional tables need to be replicated. Possible values are true or false.

Installation Process

The following are the step-by-step instructions to install the BI Interface on one database.

1. Go to the machine you are planning to work from (CA Business Service Insight source database or BI Interface machine) and copy to it the BI_Interface folder which is located under the <installation directory>/Setup folder.

   Description

   After installing CA Business Service Insight, a folder called BI_Interface can be found under the <installation directory>/Setup folder. The following folders can be found in the BI_Interface folder:

   - Class: Folder containing the java class files needed for running the batch files for generating scripts.
   - ExpImp: Folder where the export and import commands are going to be generated.
   - Generated_setup_scripts: Folder in which the setup scripts are going to be created.
   - Param_scripts: Folder where the initial scripts are stored. The scripts in this folder are used by the Build scripts batch file to create the setup scripts. Changes made to these scripts will result in changes in the generated scripts.
   - remove_archived_logs: Folder containing script files used to remove archived redo log files.
   - src: Folder containing the java class sources.
   - The following files can be found in the BI_Interface folder:
   - BuildScripts.bat: Batch file for generating the setup scripts.
   - RDLProperties.xml: XML file that contains all the relevant parameters needed for the creation of the needed scripts.
   - CreateExpImpBatch.bat: Batch file for generating the Export and Import commands with the parameter files.
   - Runsql.bat: Batch file for executing the generated scripts. Before running this batch, make sure that the scripts in the generated_setup_scripts folder have been created.
   - Undo_10.bat: Batch file for executing the Undo_10.sql script. This script removes the stream environment from the both source and target databases. It is intended for Oracle version 10 or later.
   - Undo_9.bat: Batch file for executing the Undo_9.sql script. This script removes the stream environment from both the source and the target databases. It is intended for Oracle version 9.
   - DelTBL's.bat: Batch file for deleting the imported tables and created views on the target database.

Note: In addition, each .bat file also has its UNIX equivalent file which has the same name and extension .sh.

Expected Result

You are located in the root of the location where the BI Interface installation kit exists and the files and folders exist as described in the Description column.

2. Run the CreateExpImpBatch.bat file from the BI_Interface folder.

   Description

   This operation creates all the needed SQL scripts in the generated_setup_scripts folder.

   Expected Result

   In the generated_setup_scripts folder you have 88 SQL scripts.

3. Run the BuildScripts.bat file from the BI_Interface folder.

   Description

   The BI Interface environment with all the needed processes and all the views and functions is created successfully. No errors appear in the Logs_with_errors.txt file located in the BI_Interface folder.

   Expected Result

   - The BI Interface environment with all the needed processes and all the views and functions is created successfully.
   - No errors appear in the Logs_with_errors.txt file located in the BI_Interface folder.

Post Installation Activities

After the installation process ends, check that the replication process is "alive" as follows:

1. Verify no errors occurred during installation by performing a search on the BI_Interface folder for ora-* and verify no results are found.
2. Verify that all views for the existing event types were created successfully as follows:
    a. Run the Select * from t_event_type statement and verify the result shows that a view was created for each existing event type.
    b. Run the Select * from user_tables where table_name like 'V_RD%' statement and verify the result shows that for each event type in the system, a view named V_RD_<eventtype> is created (where <eventtype> is the name of the event type for which the view is created).
    c. Run the Select * from user_tables where table_name like 'V_ID%' statement and verify the result shows that for each event type in the system, a view named V_ID_<eventtype> is created (where <eventtype> is the name of the event type for which the view is created).
3. Run the Select * from user_tables where table_name like 'V_PS%' statement and verify the result shows that V-PSL and V_PSL_EXTENDED views are created.

Chapter 4: Installation on Two Separate Databases

Installation consists of the following machines:

- CA Business Service Insight source database – The CA Business Service Insight database that needs to be replicated
- CA Business Service Insight target database – The replicated CA Business Service Insight database containing only data needed for reporting
- BI Installation – An optional machine that is used to create and run the installation scripts from instead of from the CA Business Service Insight source database

The installation process is composed of the following steps:

1. Performing prerequisites that include:
   - General Prerequisites
   - CA Business Service Insight Source Database Prerequisites
   - CA Business Service Insight Target Database Prerequisites
2. Configuration
3. Installation Process
4. Post Installation Activities

This section contains the following topics:

Prerequisites

General Prerequisites

- Create an empty database which will play the role of the CA Business Service Insight target database. This database can be created using:
  a. Oracle scripts. The tablespaces that should be created are:
     - TBS_CSL_DATA_TAB_01
     - TBS_CSL_INDEX_TAB_01
     - TBS_DASH_DATA_01
     - TBS_DASH_INDEX_01
     - TBS_LOB_DATA_01
     - TBS_OBL_DATA_01
     - TBS_OBL_INDEX_01
     - TBS_PSL_DATA_TAB_02
     - TBS_PSL_INDEX_TAB_02
     - TBS_SLA_DATA_01
     - TBS_SLA_DATA_02
     - TBS_SLA_INDEX_01
     - TBS_SLA_INDEX_02

  b. CA Business Service Insight DB Installer tool. For more information, see Creating Target Database Using DB Installer Tool (see page 57).
- If the CA Business Service Insight source and CA Business Service Insight target databases are not the same Oracle version, set or add the compatible parameter in the init.ora file on the higher database to the lower version number.

Example:

- The CA Business Service Insight source database is version 10 and the CA Business Service Insight target database is 9.
- In the init.ora file on the CA Business Service Insight source database set up the compatible parameter to equal 9.0.0.
- Verify you have enough disk space (approximately 30% of the source database size) on the machine you are working from (CA Business Service Insight source database or the BI Installation machine) for the dump file.
- Verify that the machine you are working from (CA Business Service Insight source database or the BI Installation machine) has Java Runtime Environment, version 1.5 or later, installed on it. This is required so that the batch files for installing the replication environment can be executed. Additional information on how to install and set up the Java Runtime environment can be found at the Java web site.

- Verify you have a connection from the machine you are working from (CA Business Service Insight source database or the BI Installation machine) to the CA Business Service Insight source and CA Business Service Insight target databases for sys user (e.g. sys/sys as sysdba) and check for proper connectivity to this instance using SQLPLUS (e.g. conn sys/sys@<DBNAME>).

CA Business Service Insight Source Database Prerequisites

- Verify the CA Business Service Insight source database is connected to the CA Business Service Insight target database. To verify:
  - In the tnsnames.ora file on the CA Business Service Insight source database, define a connection to the CA Business Service Insight target database.
  - From the CA Business Service Insight source database use SQLPLUS and try to connect to the CA Business Service Insight target database (e.g. conn sys/sys@<DBNAME>).
- Stop all CA Business Service Insight services if any exist. Verify that there is no mechanism that can start them up automatically.
- Set the CA Business Service Insight source database to archive log mode. For more information, see Setting CA Business Service Insight Source Database to Archive Log Mode.
- In the init.ora file on the CA Business Service Insight source database, set up the following parameters:
  - AQ_TM_PROCESSES = 1
  - JOB_QUEUE_PROCESSES = 2 or more
  - _FIRST_SPARE_PARAMETER = 50
- For Oracle10, set up the SGA_TARGET parameter to 800MB or more.
- For Oracle9, double the size of the shared_pool_size parameter, but make sure it is not less than 400MB.

Note: CA recommends that the Streams operation have 200MB allocated to the streams pool.

CA Business Service Insight Target Database Prerequisites

- In the init.ora file on the CA Business Service Insight target database, set up the following parameters:
  - AQ_TM_PROCESSES = 1
  - JOB_QUEUE_PROCESSES = 2 or more
  - _FIRST_SPARE_PARAMETER = 50
- For Oracle10, set up the SGA_TARGET parameter to 600MB or more
- For Oracle9, double the size of the shared_pool_size parameter, but make sure it is not less than 400MB

Note: CA recommends that the Streams operation have 200 MB allocated to the streams pool.

Configuration

Prior to installation, edit the RDLProperties.xml file, located under the BI_Interface folder, as follows:

- In the Source DB Users section edit the following keys according to your specifications:
  - SourceGlobalName: The global name of the source database.
  - Source_TNSNAME: The alias needed to connect to the source database from where the scripts are run.
  - SourceDBUser: An existing DB user on the source database where the CA Business Service Insight tables exist
  - SourceDBPass: The password for SourceDBUser.
  - SourceDBAdminUser: The system user of the source database.
  - SourceDBAdminPass: The password for SourceDBAdminUser.
  - SourceStreamAdminUser: This user is created by the SQL scripts, therefore a user with this name should not exist.
  - SourceStreamAdminPass: The password for SourceStreamAdminUser.
- In the Target DB Users section edit the following keys according to your specifications:
  - TargetGlobalName: The global name of the target database.
  - Target_TNSNAME: The alias needed to connect to the target database from where the scripts are run.
  - TargetDBUser: An existing DB user on the target database where the tables of the replication process are going to be updated. It should contain instances from the tables that are going to be replicated. The script creates only replication processes that keep the tables up to date.
  - CreateBookletsInterface:The password for TargetDBUser.
  - TargetDBAdminUser: The system user of the target database.
  - TargetDBAdminPass: The password for TargetDBAdminUser.
  - TargetDBRegularUser: This user is created. This user is granted read-only access to the replicated data and the created views, excluding: full T_PSL interface, T_PSL, T_RAW_DATA, T_SLALOM_OUTPUTS, and T_INTERMEDIATE_DATA.
  - TargetDBRegularPass: The password for TargetDBRegularUser.
  - TargetStreamAdminUser: This user is created by the SQL scripts, therefore a user with this name should not exist.
  - TargetStreamAdminPass: The password for TargetStreamAdminUser.
  - FilePath: Specifies the location of the database file for new table space.

- In the Tables to be replicated and interfaces to be created section edit the following keys according to your specifications:
  - Replication: Specifies whether installation is to be on two separate databases connected using streams technology (true) or on the original database (false). In this type of installation set the parameter to true.
  - ReplicatedTables_PSL: List of tables needed to be replicated for the T_PSL interface.
  - CreatePSLInterface: Specifies whether the T_PSL interface should be created. Possible values are true or false.
  - ReplicatedTables_RAW_DATA: List of tables needed to be replicated for the T_RAW_DATA interface.
  - CreateRAW_DATAInterface: Specifies whether the T_RAW_DATA interface should be created. Possible values are true or false.
  - ReplicatedTables_SLALOM_OUTPUTS: List of tables needed to be replicated for the T_SLALOM_OUTPUTS interface.
  - CreateSLALOM_OUTPUTSInterface: Specifies whether the T_SLALOM_OUTPUTS interface should be created. Possible values are true or false.
  - ReplicatedTables_Booklets: List of tables needed to be replicated for the Booklet mapping interface.
  - CreateBookletsInterface: Specifies whether the Booklet mapping interface should be created. Possible values are true or false.
  - AdditionalTables: List of any additional tables that need to be included in the replication process.
  - AdditionalTablesInterface: Specifies whether the Additional tables need to be replicated. Possible values are true or false.

Installation Process

The following are the step-by-step instructions to install the BI Interface on two seperate databases.

1. Go to the machine you are planning to work from (CA Business Service Insight source database or BI Interface machine) and copy to it the BI_Interface folder which is located under the <installation directory>/Setup folder.

   Description

   After installing CA Business Service Insight, a folder called BI_Interface can be found under the <installation directory>/Setup folder.

   The following folders can be found in the BI_Interface folder:

   - Class: Folder containing the java class files needed for running the batch files for generating scripts.
   - ExpImp: Folder where the export and import commands are going to be generated.
   - Generated_setup_scripts: Folder in which the setup scripts are going to be created.
   - Param_scripts: Folder where the initial scripts are stored. The scripts in this folder are used by the Build scripts batch file to create the setup scripts. Changes made to these scripts will result in changes in the generated scripts.
   - remove_archived_logs: Folder containing script files used to remove archived redo log files.
   - src: Folder containing the java class sources.
   - The following files can be found in the BI_Interface folder:
   - BuildScripts.bat: Batch file for generating the setup scripts.
   - RDLProperties.xml: XML file that contains all the relevant parameters needed for the creation of the needed scripts.
   - CreateExpImpBatch.bat: Batch file for generating the Export and Import commands with the parameter files.
   - Runsql.bat: Batch file for executing the generated scripts. Before running this batch, make sure that the scripts in the generated_setup_scripts folder have been created.
   - Undo_10.bat: Batch file for executing the Undo_10.sql script. This script removes the stream environment from the both source and target databases. It is intended for Oracle version 10 or later.
   - Undo_9.bat: Batch file for executing the Undo_9.sql script. This script removes the stream environment from both the source and the target databases. It is intended for Oracle version 9.

- DelTBL's.bat: Batch file for deleting the imported tables and created views on the target database.

Note: In addition, each .bat file also has its UNIX equivalent file which has the same name and extension .sh.

Expected Result

You are located in the root of the location where the BI Interface installation kit exists and the files and folders exist as described in the Description column.

2. Run the CreateExpImpBatch.bat file from the BI_Interface folder.

   Description

   To set up the Streams environment, you must first instantiate the tables that are going to be replicated. This operation creates the ExportTables.bat, ImportTables.bat, ExpParameterFile.txt, ImpParameterFile.txt, and PREP_INST.sql files. The CreateExpImpBatch.bat file uses the RDLProperties.xml file to read the data for the source and target databases for the tables that need to be replicated and for the Users that need to be created.

   Expected Result

   In the ExpImp folder the following files exist:

   - ExportTables.bat
   - ImportTables.bat
   - ExpParameterFile.txt
   - ImpParameterFile.txt
   - PREP_INST.sql

3. Run the BuildScripts.bat file from the BI_Interface folder.

   Description

   This operation creates all the needed SQL scripts in the generated_setup_scripts folder.

   Expected Result

   In the generated_setup_scripts folder you have 88 SQL scripts.

4. Run ExportTable.bat from ExpImp folder.

   Description

   This operation performs an export of the tables that are written in the ExpParameterFile.txt file to the RDLExportedTables.dmp file.

   Expected Result

   - The RDLExportedTables.dmp file is created in the ExpImp folder.
   - No errors appear in the ExportTable.log file located in the ExpImp folder.

5. Run ImportTables.bat from ExpImp folder.

Description

This operation uses the RDLExportedTables.dmp file for importing to the target database. The ImportTables.bat file and other settings for performing the import are written in the ImpParameterFile.txt file.

Expected Result

The import process completed successfully and no errors appear in the ImportTables.log file located in the ExpImp folder.

6. Run the runsql.bat file from the BI_Interface folder.

Description

This operation creates the Streams environment with all the needed processes and all the views and functions.

Expected Result

- The Streams environment with all the needed processes and all the views and functions is created successfully.
- No errors appear in the Logs_with_errors.txt file located in the BI_Interface folder.

Post Installation Activities

After the installation process ends, check that the replication process is "alive" as follows:

1. Verify no errors occurred during installation as follows:
   a. Perform a search on the BI_Interface folder for ora-* and verify no results are found.
   b. On the CA Business Service Insight target database, run the select*from dba_apply_error and verify no results are found.
2. Verify the replication mechanism is operating properly as follows:
   a. Make a change in a table in the CA Business Service Insight source database:

```
update t_users set user_modify_date=sysdate where user_id=0;
commit;
```

   b. Wait several minutes.
   c. Check in the CA Business Service Insight target database that you see the updated data, for example:

```
select user_modify_date from t_users where user_id=0;
```

Chapter 5: Creating Target Database Using DB Installer Tool

Note: This section is only relevant when configuring the BI Interface for installation on two separate databases.

To create the CA Business Service Insight target database using the CA Business Service Insight DB Installer tool:

1. Copy the InstallDB folder from the CA Business Service Insight package /SetupDB folder.
2. Edit the start_parameters.ini file according to the start_parameters_readme.txt file.
3. Run the Main.bat file (for Windows) or Main.sh (for Unix).
4. Recreate the CA Business Service Insight user as follows:

```
DROP USER OBLICORE CASCADE;

CREATE USER OBLICORE
IDENTIFIED BY OBLICORE
DEFAULT TABLESPACE TBS_SLA_DATA_01
TEMPORARY TABLESPACE TEMP
ACCOUNT UNLOCK
/
GRANT  ALTER SESSION TO OBLICORE
/
GRANT  CONNECT TO OBLICORE
/
GRANT  CREATE DATABASE LINK TO OBLICORE
/
GRANT  CREATE PROCEDURE TO OBLICORE
/
GRANT  CREATE PUBLIC SYNONYM TO OBLICORE
/
GRANT  CREATE SEQUENCE TO OBLICORE
/
GRANT  CREATE SYNONYM TO OBLICORE
/
GRANT  CREATE TABLE TO OBLICORE
/
GRANT  CREATE TRIGGER TO OBLICORE
/
GRANT  CREATE TYPE TO OBLICORE
/
GRANT  CREATE VIEW TO OBLICORE
/
GRANT  DROP PUBLIC SYNONYM TO OBLICORE
/
GRANT  EXECUTE ANY PROCEDURE TO OBLICORE
/
GRANT  EXECUTE ON DBMS_LOCK TO OBLICORE
/
GRANT  SELECT ON V_$SESSION TO OBLICORE
```

```
/
GRANT  SELECT ANY TABLE TO OBLICORE
/
GRANT  UNLIMITED TABLESPACE TO OBLICORE
/
GRANT SELECT ANY DICTIONARY TO OBLICORE
/
```

Chapter 6: Setting Source Database to Archive Log Mode

Note: This section is only relevant when configuring the BI Interface for installation on two separate databases.

To switch the CA Business Service Insight source database archiving mode, use the ALTER DATABASE statement with the ARCHIVELOG or NOARCHIVELOG option. The following steps indicate how to switch the CA Business Service Insight source database archiving mode from NOARCHIVELOG to ARCHIVELOG:

1. Shut down the database instance using the SHUTDOWN command.

   An open database must first be closed and any associated instances shut down before you can switch the database's archiving mode. You cannot disable archiving if any data files need media recovery.

2. Back up the database.

   Before making any major change to a database, always back up the database to protect against any problems. This will be your final backup of the database in NOARCHIVELOG mode and can be used if something goes wrong while trying to change to ARCHIVELOG mode. See Oracle9i User-Managed Backup and Recovery Guide or Oracle9i Recovery Manager User's Guide for further information on backup and recovery.

3. Edit the initialization parameter file to include initialization parameters specifying whether automatic archiving is enabled (see "Enabling Automatic Archiving") and the destinations for the archive log files (see "Specifying Archive Destinations").

   ```
   LOG_ARCHIVE_START=TRUE
   LOG_ARCHIVE_DEST = '/disk1/arc'
   ```

4. Start a new instance and mount, but do not open, the database using the STARTUP MOUNT command.

   Note: To enable or disable archiving, the database must be mounted but not open.

5. Switch the database's archiving mode. Then open the database for normal operations as follows:

   ```
   ALTER DATABASE [NO]ARCHIVELOG;
   ALTER DATABASE OPEN;
   ```

6. Shut down the database using the SHUTDOWN IMMEDIATE command.

7. Back up the database.

Chapter 7: Removing the Stream Environment

Note: This section is only relevant when configuring the BI Interface for installation on two separate databases.

To remove the streaming configuration of both target and source databases, run the UnDo_10.bat or UnDo_9.bat file, depending on your Oracle version.

To remove tables and views from the target database, run the DelTBL's.bat file.

Chapter 8: Reconfiguration

For each interface that exists in the CA Business Service Insight database, set to false the value of the parameter that specifies whether the interface should be created (e.g. CreateINTERMEDIATE_DATAInterface, CreateRAW_DATAInterface, and so on).

Chapter 9: Troubleshooting the Stream Environment

Note: This section is only relevant when configuring the BI Interface for installation on two separate databases.

If data replication does not work, please consult the following troubleshooting list:

- To verify that the capture process is working, check that the status in table DBA_CAPTURE on source database is enable. If the status is disabled, an error will appear in the ERROR_MESSAGE field of this table
- To verify that instantiation is properly set on the replicated tables, check that all replicated tables are DBA_CAPTURE_PREPARED_TABLES on the source database and that they are in the DBA_APPLY_INSTANTIATED_OBJECTS on the target database
- To verify that the propagation process is working, check that the status in table DBA_PROPAGATION on source database is enable. If the status is disabled, an error will appear in ERROR_MESSAGE field of this table
- To verify that the apply process is working, check that the status in table DBA_APPLY on the target database is enabled. If the status is disabled, an error will appear in the ERROR_MESSAGE field of this table and in the DBA_APPLY_ERROR table on the target database
- To verify that the views show proper data, check that no errors exist in the t_raw_data_fields_log table on the CA Business Service Insight target database

Chapter 10: Introduction

This document contains the release notes for the Business Intelligence Interface. The following chapters describe the release in detail and provide other information that supplements the main documentation.

This release improves quality, reliability, performance and adds new functionality.

Chapter 11: Installation and Upgrade

For detailed installation instructions, see the CA Business Service Insight BI Interface Guide document.

Chapter 12: Miscellaneous Enhancements

No new enhancement were made to the BI interface in this release.

Chapter 13: Known Issues

- There is a known issue where after installing/upgrading the BI interface using the “Oblicore” Oracle database user (privileges), errors occur that prevent a successful installation.

  To resolve:

  1. Run the oblicore_privs.sql on the BI interface database instance using a user that has SYSDBA priviliges. This file is located under the migration root directory (for example: C:\Program Files\Oblicore\Database Installation\Migration). You can also get the script from the Installation CD under the SetupDB\Migration folder.
  2. Run the runsql.bat file to begin installation.

- Views of entities that have an entity name larger than 30 bytes cannot be created by the installation.