User Guide

CUSTOMER

Document Version: 1.0 – 2015-07-23

SAP Landscape Transformation Replication Server

Setting up replication of INDX tables

Typographic Conventions

Type Style	Description
Example	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
Example	Emphasized words or expressions.
EXAMPLE	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE	Keys on the keyboard, for example, F2 or ENTER.

Document History

Version	Date	Change
1.0	2015-07-23	Initial Version

Table of Contents

1 Use Case

You are using SAP LT Replication Server to replicate data from an SAP source system to an SAP HANA database.

You would like to replicate INDX-like tables (cluster databases). The typical examples of these tables are STXL, PCL2, INDX.

1.1 Restrictions

SAP System Landscape Optimization team provides the pilot solution to replicate INDX-like tables in the latest releases of SAP SLT Replication Server (as of DMIS 2011 SP6). As long as it is not officially released as a feature, not all aspects might work as expected e.g. Replication Logging of such tables. We are continuously improving the usability and functionality of the INDX-like table replication. Please refer to the corresponding central note for the corresponding SLT release or open the support message for component HAN-DP-LTR.

To enable the declustering and replication of the tables with compressed fields (INDX tables) following prerequisites are necessary:

- SAP DMIS 2011 SP6 or higher is installed on the central SLT system
- Following notes should be installed to enable the INDX-like tables replication (see the description of each note regarding the systems where it should be installed)
    - 1964064
    - 2001386
    - 2042447
    - 2140471
- We recommend to use the latest DMIS release with the latest correction notes installed.

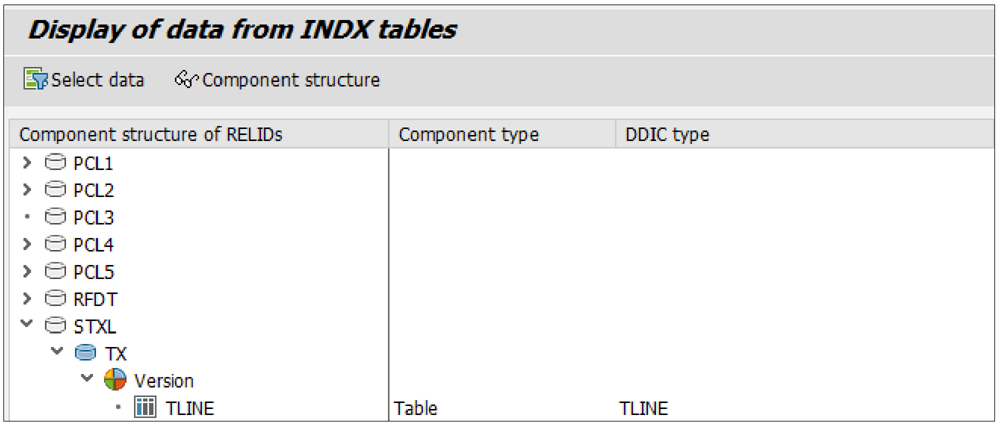
1.2 Cluster databases (INDX-like tables)

Cluster databases are the relational database tables in ABAP Dictionary that can be used to store the data clusters of arbitrary structure. The data structure to be stored in cluster is defined by the application and data are stored in a special cluster field in the INDX-like table in the compressed form.

The INDX-like table have predefined structure in the ABAP Dictionary where several transparent fields store the data that allows determining the structure of the cluster field (e.g. cluster area ID RELID, cluster name, cluster block counter SRTF2 etc.). For more information on cluster databases please refer to the SAP Help http://help.sap.com/saphelp_470/helpdata/en/fc/eb3c05358411d1829f0000e829fbfe/content.htm .

1.3 Declustering logic

Based on experience collected in data conversion and data migration projects for our customers SAP SLT Replication Server provides the tools and knowledgebase for declustering the cluster database tables. You can find the knowledgebase delivered by SAP in the transaction CNV_INDX_OVERVIEW.



Component structure of RELIDs	Component type	DDIC type
PCL1		
PCL2		
PCL3		
PCL4		
PCL5		
RFDT		
STXL		
TX		
Version		
TLINE	Table	TLINE

Pic.1.1. Transaction CNV_INDX_OVERVIEW.

During the configuration, the necessary declustering procedures as well as data structure mappings should be defined to enable the data declustering on the fly during the replication.

2 Configuration

2.1 Overview

The setup process consists of three phases.

1. Setup declustering logic
   - Define necessary types of data for extraction
   - Generate declustering routines

2. Define Target table structure
   - Specify the location of the target structure definition
   - Describe key fields structure
   - Define declustered data structure

3. Define Replication settings
   - Define target table for the given configuration
   - Key field mapping
   - Activate key structure definition

2.2 Setup declustering logic

In this step, the declustering rountines are generated on the SLT system based on the provided definition customizing. It is possible to generate the modules based on the database cluster definition in the central system or in the source system. There are following considerations should be taken into account

Build declustering rountines based on the definition in the source system	Build declustering rountines based on the definition in the central SLT system
• Table definition doesn't exist on the SLT system and can't be created on the SLT system • One SLT system is used for many different configurations where the database cluster table definition may differ	• Table definition exists on the SLT system (or can be created) • Only one(few) configurations are used and there's no different structure definition for the database cluster table between different sources

⚠ Depending on the preferred strategy, you should perform the steps in 2.2.1 on the source or central SLT system.

⚠ As an example in this document, we will setup replication of the table **STXL**. Please note that as on DMIS 2011 SP8 the basic settings for cluster structure (p.2.2.1.) are provided in standard solution. In this case you can process with p.2.2.2.

2.2.1 Provide information about the database cluster table structure

1. Start transaction CNV_INDX_KNOWLEDGE, Select INDX tables, switch to Edit mode, and create entry for the table **STXL** by clicking the button New Entries in the menu bar. Following data to be provided:
   - Create entry for the table (here STXL)
   - Provide name of the field where the cluster area ID is stored (**RELID**)
   - Provide the cluster line number field name (SRTF2).This field is used as the counter for the different cluster segments stored in the different table lines.
   - Some database cluster tables allow versioning for the cluster area ID, in this case the versioning information is stored in the corresponding field. In this case provide the name of the field used for versioning (usually VERSN). It is not relevant for the table STXL.
   - Save and return to the previous screen

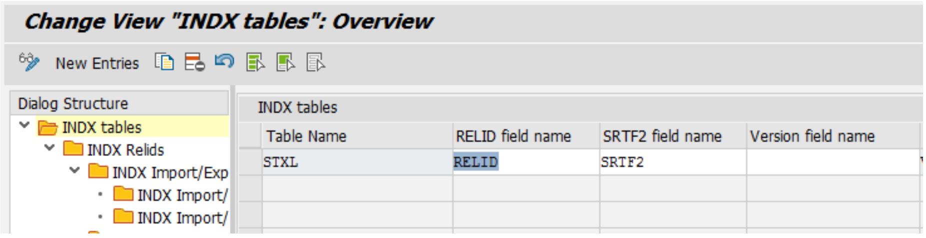


Table Name	RELID field name	SRTF2 field name	Version field name
STXL	RELID	SRTF2	

Pic.2.2.1.1. Define INDX-like table structure in the transaction CNV_INDX_KNOWLEDGE.

2. Select the newly created record and double click on tree node "INDX Relids"
   - Provide the cluster areas(identified by RELID) that you want to transfer from the INDX-like table (for texts stored in table STXL it is: **TX**).
   - Save and return to the previous screen

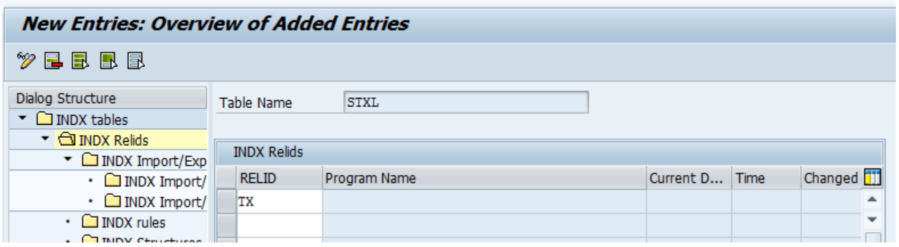


Table Name STXL

RELID	Program Name	Current D...	Time	Changed
TX				

Pic.2.2.1.2. Define list of cluster areas that you would like to decluster and replicate.

3. Select the newly created record for the cluster area ID and double click on tree node "INDX Structures". Provide here the structure of the cluster data components. The list of data components is specific for each database cluster and usually integrated into the application logic. Note that cluster structure may change between the versions, therefore you can provide here the different structure definition for the component versions/variants.

   ⚠ You should define the same structure that is defined in the corresponding application. Please involve application expert if you require any help on the declustering of the table that is not yet provided in the SLT knowledgebase.

   The structure of the data component can be defined by an ABAP dictionary element (for cluster area ID TX of the table STXL it is **TLINE**)

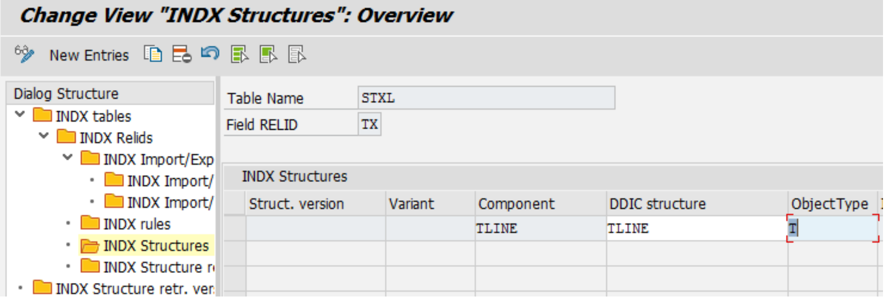


Table Name STXL

Field RELID TX

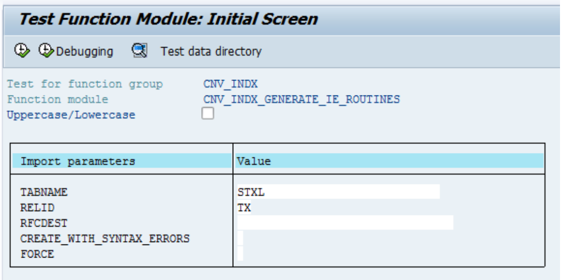
Struct. version	Variant	Component	DDIC structure	ObjectType
		TLINE	TLINE	T

Pic.2.2.1.3. Define component structures of the corresponding cluster area ID.

2.2.2 Generate rountines for the data declustering

The declustering takes place on the central system therefore this step should be performed in the central system. Should the steps 2.2.1. be performed in the source system, please provide the RFC connection details to the corresponding source system in the field RFCDEST.

In transaction SE37 execute the function module CNV_INDX_GENERATE_IE_ROUTINES for each cluster area ID RELID relevant for the replication



Test for function group CNV_INDX

Function module CNV_INDX_GENERATE_IE_ROUTINES

Uppercase/Lowercase □

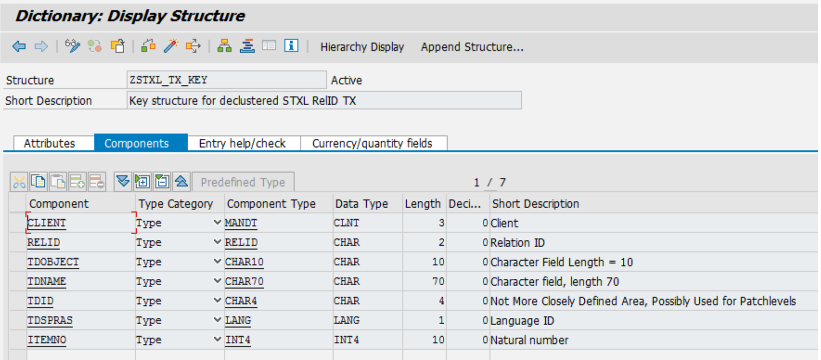
Import parameters	Value
TABNAME	STXL
RELID	TX
RFCDEST	
CREATE_WITH_SYNTAX_ERRORS	
FORCE	

Pic.2.2.2.1. Generate the declustering rountines for each cluster area ID.

2.3 Define Target table structure

This step should be performed on the SLT system,

1. Please create the structure that describes the key fields of the target table. Open transaction SE11 and create the structure. The list of key fields should at least include the key fields of the cluster database table.



Structure ZSTXL_TX_KEY Active

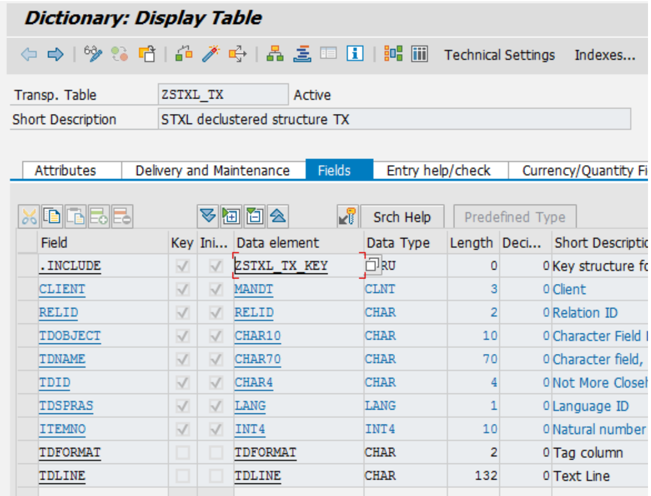
Short Description Key structure for declustered STXL RelID TX

Component	Type Category	Component Type	Data Type	Length	Deci...	Short Description
CLIENT	Type	MANDT	CLNT	3	0	Client
RELID	Type	RELID	CHAR	2	0	Relation ID
TDOBJECT	Type	CHAR10	CHAR	10	0	Character Field Length = 10
TDNAME	Type	CHAR70	CHAR	70	0	Character field, length 70
TDID	Type	CHAR4	CHAR	4	0	Not More Closely Defined Area, Possibly Used for Patchlevels
TDSPRAS	Type	LANG	LANG	1	0	Language ID
ITEMNO	Type	INT4	INT4	10	0	Natural number

Pic.2.3.1. Key structure of the target table.

⚠ Please note that declustered table usually contain much more records than the cluster database therefore please adjust the type of the counter field accordingly (in current example field ITEMNO has type INT4).

2. Please create the table that describes the structure of the target table. Open transaction SE11 and create the table. You can reuse the the structure created in the previous step for the definition of the table key fields.



Dictionary: Display Table

Transp. Table	ZSTXL_TX	Active
Short Description	STXL declustered structure TX	

Field	Key	Ini...	Data element	Data Type	Length	Deci...	Short Descriptio
.INCLUDE	✓	✓	ZSTXL_TX_KEY	RU	0	0	Key structure fo
CLIENT	✓	✓	MANDT	CLNT	3	0	Client
RELID	✓	✓	RELID	CHAR	2	0	Relation ID
TDOBJECT	✓	✓	CHAR10	CHAR	10	0	Character Field I
TDNAME	✓	✓	CHAR70	CHAR	70	0	Character field,
TDID	✓	✓	CHAR4	CHAR	4	0	Not More Closel
TDSPRAS	✓	✓	LANG	LANG	1	0	Language ID
ITEMNO	✓	✓	INT4	INT4	10	0	Natural number
TDFORMAT	□	□	TDFORMAT	CHAR	2	0	Tag column
TDLINE	□	□	TDLINE	CHAR	132	0	Text Line

Pic.2.3.2. Target table definition on SLT system.

2.4 Define replication settings

Following settings are necessary to map the fields and declustered data from the source table into the target transparent table.

2.4.1 Provide information about the target table structure

In the SLT system open the table IUUC_INDX_OBJECT in transaction SE16. For each configuration where you setup replication of the cluster database table create the following record

- Mass Transfer ID of SLT Configuration
- Table name (here **STXL**)
- Cluster area ID (RELID, here **TX**)
- Target table in field TARGET_TABLE field (here: **TLINE**, see 2.3)
  Target table name for declustering the data of this area ID
- Target table name in TARGET_TAB_NAME (here **ZSTXL_TX**)
  You can define a deviating table name where the data of this area ID will be declustered to. Thus it is possible to split the data of the database cluster table into the different tables during declustering
- Cluster DDIC structure name in STRUCT_NAME (here: **TLINE**)
  The structure provided here will be used as the definition of the structure where the cluster data will be declustered to. Please use this field if the replication is setup on the SLT system (see 2.2 as well)

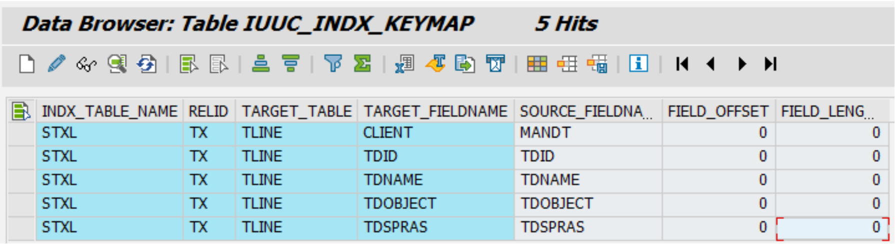
- If you prefer using a table from the source system instead of a structure on the SLT system to define the structure of declustered data(in the field STRUCT_NAME), please enter its name in the field SOURCE_TAB_NAME

Please note that one database cluster table may consist different segments that could not be declustered into the same data structure. Therefore settings in this table define the target table properties(table structure and table name if you would like to replicate the data into a table with a different name) per data areaID and corresponding data structure of the database cluster table,

2.4.2 Define key field mapping for the declustered transparent table

In order to decluster data into the transparent table it is necessary to provide the key field mapping between the database cluster table and the target transparent table. These settings should be maintained in the table IUUC_INDX_KEYMAP:

- Use the same logical table name as provided in the 2.4.1 TARGET_TABLE field (here TLINE, see slide 13, p.5 for the TARGET_TABLE)
- For each key field provide a corresponding target structure field (fields SOURCE_FIELDNAME and TARGET_FIELDNAME correspondingly)



Data Browser: Table IUUC_INDX_KEYMAP 5 Hits

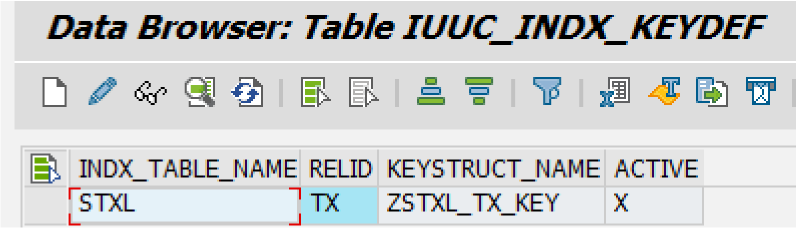
INDX_TABLE_NAME	RELID	TARGET_TABLE	TARGET_FIELDNAME	SOURCE_FIELDNA…	FIELD_OFFSET	FIELD_LENG…
STXL	TX	TLINE	CLIENT	MANDT	0	0
STXL	TX	TLINE	TDID	TDID	0	0
STXL	TX	TLINE	TDNAME	TDNAME	0	0
STXL	TX	TLINE	TDOBJECT	TDOBJECT	0	0
STXL	TX	TLINE	TDSPRAS	TDSPRAS	0	0

Pic.2.4.2. Key field mapping in the table IUUC_INDX_KEYMAP.

2.4.3 Assign Key structure definition

Provide key field structure for declustering of each area ID:

- Provide table name
- Cluster area ID (RELID)
- Provide the name of the structure describing the key fields
- Set flag Active to X



Data Browser: Table IUUC_INDX_KEYDEF

INDX_TABLE_NAME	RELID	KEYSTRUCT_NAME	ACTIVE
STXL	TX	ZSTXL_TX_KEY	X

Pic.2.4.3. Activation of the Key structure IUUC_INDX_KEYDEF.

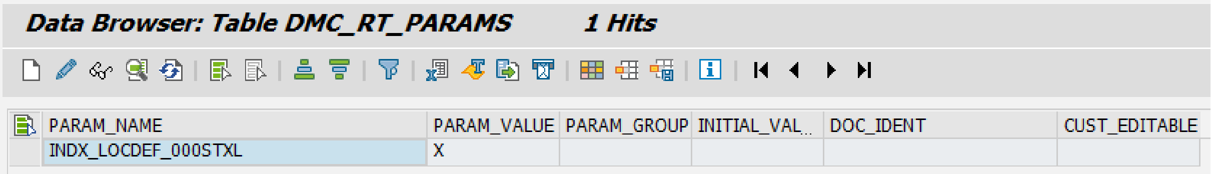
2.4.4 SLT configuration and load start

2.4.4.1 Define the source for the declustering data structure definitions

This step should only be performed if definition of the step 2.2. is performed on the SLT system.

In the table DMC_RT_PARAMS in transaction se16 and create the following record for the corresponding database cluster table (here **STXL**):

- PARAM_NAME should be built of three parts
  - Prefix INDX_LOCDEF_
  - MT_ID or 000 if the settings are relevant for all configurations
  - Table name (here **STXL**)
- Set PARAM_VALUE to X



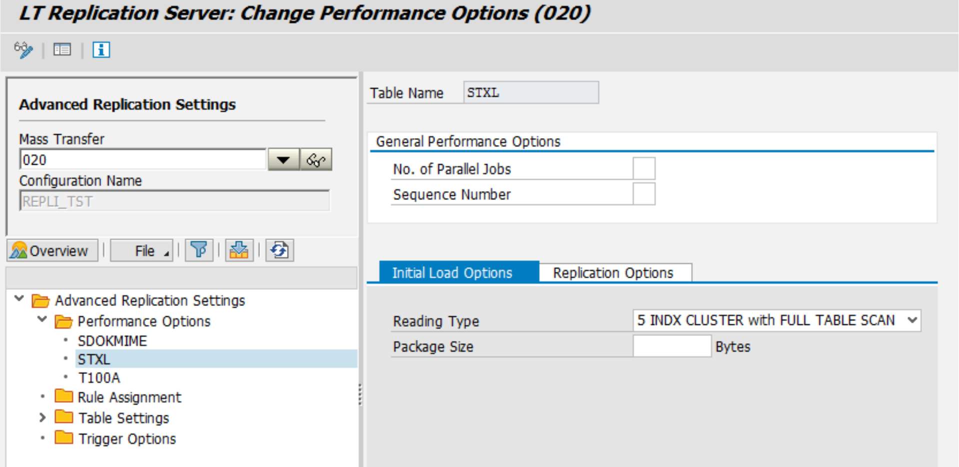
Data Browser: Table DMC_RT_PARAMS 1 Hits

PARAM_NAME	PARAM_VALUE	PARAM_GROUP	INITIAL_VAL..	DOC_IDENT	CUST_EDITABLE
INDX_LOCDEF_000STXL	X				

Pic.2.4.4.1. Activation of the local structure definition.

2.4.4.2 Use cluster reading type for the data migration

Define reading type for the database cluster table initial load. We recommend to use reading type 5 for the initial load(could be defined in transaction LTRS)



LT Replication Server: Change Performance Options (020)

Advanced Replication Settings

Mass Transfer: 020

Configuration Name: REPLI_TST

Table Name: STXL

General Performance Options

No. of Parallel Jobs:

Sequence Number:

Reading Type: 5 INDX CLUSTER with FULL TABLE SCAN

Package Size: Bytes

Pic.2.4.4.2. Reading type for the database cluster table.

2.4.4.3 Start load/replication

Use standard data provisioning functionality in transaction LTRC to start the table load/replication.

www.sap.com