

White Paper

ABAP Post-Copy Automation

Document Version: 2.0 – 2015-05-04

CUSTOMER

Conversion of Logical System Names

**BDLS Tasks starting with SAP Note 2039075 - System Copy: Task Content
8. Improvements**



Typographic Conventions

Type Style	Description
<i>Example</i>	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	2015-02-20	Initial Version
2.0	2015-05-04	Minor changes

Table of Contents

1	Introduction	5
1.1	Logical System Names	5
1.2	Logical System Name Conversion with BDLS	6
1.3	Logical System Name Conversion with Post-Copy Automation.....	7
2	Post-Copy Automation - BDLS Tasks	9
2.1	Overview	9
2.2	Prerequisites.....	10
2.3	Procedure.....	10
3	Post-Copy Automation - Optimized BDLS Task.....	13
3.1	Overview of Analysis Options	13
3.2	History	14
3.3	Execution	14
3.3.1	Option A: Execute report for table analysis before system copy in source system	14
3.3.2	Option B: Run table analysis together with BDLS conversion	16
3.3.3	Option C: Execute table analysis prior to executing BDLS conversion.....	16
4	Best Practices.....	20
5	Conversion Log Details.....	21
6	More Information	22

1 Introduction

1.1 Logical System Names

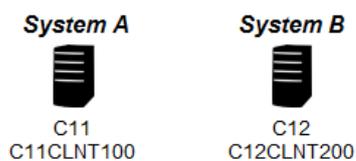
For several use cases, you have to assign unique IDs to systems participating in a distributed environment.

For example:

- Communication partners in Application Linking and Enabling (ALE) scenarios such as Interface technology developed by SAP to loosely connect application systems (SAP and external systems) with one another.
- Communication partners in an SAP Business Warehouse scenario such as connecting SAP BW systems and SAP BW source systems.
- Communication partners in central user administration.

For systems based on SAP NetWeaver ABAP, logical system names are defined as unique ID.

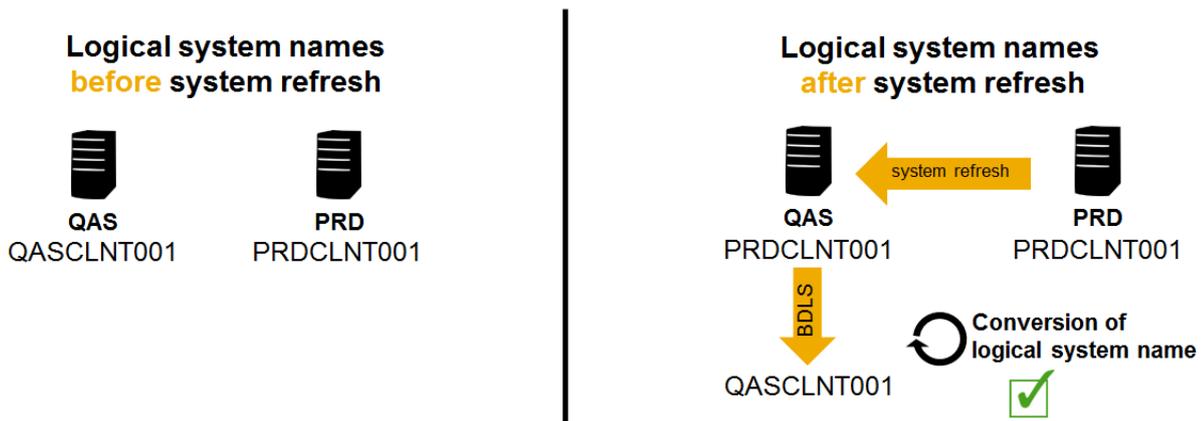
You assign a name explicitly to one client in an SAP system. For example:



Participating System	Unique ID
System A (SAPSID: C11), client 100	C11CLNT200
System B (SAPSID: C12), client 200	C12CLNT200

1.2 Logical System Name Conversion with BDLS

In case of a system copy or a system refresh, logical system names get duplicated and you have to convert them afterwards, to end up with unique IDs again. To convert one or more logical system names to new or existing logical names, you use transaction `BDLS`. Transaction `BDLS` determines all relevant tables (application tables, master data tables, control data tables) using specific logical system names and allows converting the corresponding table entries (field values) to a different logical system name.



`BDLS` performs the following steps:

1. It determines all active, transparent database tables, whose fields have references to following domains:
LOGSYS, EDI_PARNUM
2. It converts corresponding field values to new logical system names
3. It updates the database

But logical system name conversion usually takes a long time, as the standard conversion program used by `BDLS` scans all tables in your system and converts the corresponding logical system names – especially in case your system uses many logical system names or has very huge tables (such as in the range of billion table entries), this most probably results in a long-running conversion task. To improve this procedure, you can use ABAP PCA for `BDLS`.

For more information about transaction `BDLS`, search for *Conversion of Logical Systems* within your SAP NetWeaver release on SAP Help Portal at <http://help.sap.com>.

1.3 Logical System Name Conversion with Post-Copy Automation

As an enhancement to transaction `BDLS`, ABAP post-copy automation (PCA) offers automated procedures that improve the performance and reduce the duration of the conversion. For this, PCA offers 3 execution modes in the `BDLS` conversion task:

- Job execution mode 1: *Single Job Execution*
- Job execution mode 2: *Concurrent Jobs based on Table Name*
- Job execution mode 3: *Concurrent Jobs based on Table Entry*

Conversion of Logical System Names

Logical System Name Mapping: OLDCLNT000 to NEWCLNT000

Tables to be Converted: [Empty field]

Number of Entries per Commit: 100.000

Job Execution Mode

- Single Job Execution
- Concurrent Jobs based on Table Name
- Concurrent Jobs based on Table Entry

Use Analysis of Logical System Names Task

Use Existing Table Analysis (RBDLS_CHECK)

Analysis outdated after (days): 180

Perform Table Analysis at Execution

Number of Parallel Process: 5

RFC Server Group: [Empty field]

Release Jobs

Maintain RFC Destination

With ABAP PCA for `BDLS`, you can improve the procedure as follows:

- Define more than one conversion name pair
- Execute concurrent jobs for performance improvements
- Execute the `BDLS` conversion in other clients than executing client
- Execute the `BDLS` conversion in multiple clients
- Use built-in checks to verify the parameter consistency
- Execute the conversion with background jobs

For more information, see [Post-Copy Automation - `BDLS` Tasks](#).

If the achieved runtime does not suffice your requirements, you can further optimize the process by reducing the amount of tables for conversion. For this, you select job execution mode 3, to process according to table entry and get several options for running a detailed analysis to isolate tables affected by the logical system name change:

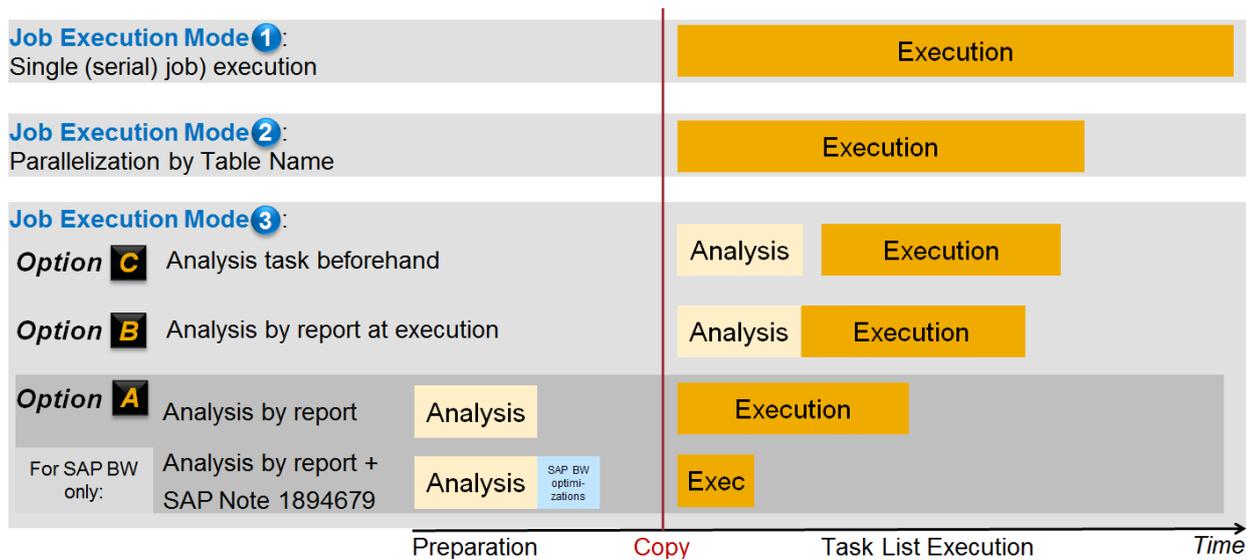
- Option A: You execute a specific report for the table analysis before system copy in the source system.
- Option B: You run the table analysis at the execution of the configuration task for `BDLS` conversion.
- Option C: You execute a specific configuration task for table analysis prior to executing the configuration task for `BDLS` conversion.

Using job execution mode 3, you have the following additional benefits:

- Better planning possibilities of the system copy project
- Reduced amount of tables for conversion
- Optimal parallelization of conversion
- Optimized job distribution

For more information, see [Post-Copy Automation - Optimized BDLS Task](#).

Rough execution comparison of the possible options:



This paper describes the underlying concept, how the actual conversion works, provides further details on the different analysis options and gives recommendations on how to use the provided features.

i Note

The BDLS tasks for analysis and conversion are part of ABAP PCA, which itself is part of SAP Landscape Virtualization Management, enterprise edition software. SAP Landscape Virtualization Management is an add-on to SAP NetWeaver 7.3 and SAP NetWeaver 7.4.

To use ABAP PCA, you must own an SAP Landscape Virtualization Management, enterprise edition license. For more information, see the Installation and Configuration Guides for ABAP PCA on SAP Service Marketplace at <http://service.sap.com/instlvm>.

For more information about using the PCA content to configure extensive technical scenarios automatically, see the ABAP post-copy automation installation and configuration guides on SAP Service Marketplace at <http://service.sap.com/instlvm>.

2 Post-Copy Automation - BDLS Tasks

2.1 Overview

PCA offers the BDLS task `CL_STCT_SC_CONV_BDLS` to convert the logical system names.

The configuration task `CL_STCT_SC_CONV_BDLS` uses the programs and the logic used in transaction `BDLS` but contains several improvements compared to transaction `BDLS`. The tables are processed using program `RBDLSMAP2`.

Note

Depending on your release level, program `RBDLSMAP` is used instead. For more information, see SAP Note [1714286](#) - BDLS: Prevention of various terminations.

Using this configuration task, you have the possibility to define more than one conversion name pair. It is possible to execute concurrent jobs for performance improvements and to execute the BDLS conversion in other clients than executing client. The conversion execution is done with background jobs.

The conversion is divided into the following phases:

- Standard table conversion
Standard tables are processed using program `RBDLSMAP2`. The conversion jobs are executed in parallel.

Recommendation

To speed up the conversion, configure as many background work processes as possible.

- Special table conversion routines
Special tables (exceptional objects), for example very large tables or application specific tables which need special BDLS treatment, are processed at the end using an SAP-specific program and a customer-specific program. Standard SAP tables are defined in table `BDLSEXT` and customer-specific tables in table `BDLSEXZ`. With that you have the option to exclude large tables or to start a program before or after conversion. Special tables (exceptional objects) are also then processed using program `RBDLSMAP2`.
For more information, see SAP Note [932032](#) - BDLS: Special conversion for large tables.

By default, conversions are carried out in the current client, that is, for all client-specific and cross-client table entries.

By configuring RFC destinations to the respective clients in the local system, it is also possible to execute remote client conversions in the local system. These follow-on client conversions are done only for client-specific tables.

Note

Special table (exceptional object) conversions are also executed per client when performing multiple client conversions. Therefore, customer-specific programs must be modified to avoid being executed multiple times.

Logical system name pairs are processed one pair after another during multi-pair conversions. Similarly, in multi-client conversions, clients are also processed one after the other. When performing multiple client conversions, the client-specific conversion and cross-client table conversion are each done only once, that the amount of tables for conversion is reduced. The task also has built-in checks that verify the parameter consistency.

2.2 Prerequisites

Use a remote function call (RFC) destination of type 3 for the target client with an equivalent administrator user that is used to execute the post-copy automation tool when performing multiple client conversion.

2.3 Procedure

1. Choose  (Fill Parameters) in the *Parameter* column of task CL_STCT_SC_CONV_BDLS.

Figure 1-CL_STCT_SC_CONV_BDLS User Interface

2. Provide the following information:

Field	Description
<i>Logical System Name Mapping</i>	To define the logical system name conversion pairs, enter an old and a new logical system name. To define multiple pairs, specify the first pair, choose  (Multiple selection) and specify the remaining pairs.
<i>Tables to be Converted</i>	Specify tables for conversion and target specific tables in the database: <ul style="list-style-type: none"> • To convert the names for all related tables, leave the field empty. • To convert the names for selected tables, choose  (Multiple selection) and enter the required data. • To exclude the names for specific tables, choose  (Multiple selection) and enter the required data. • To deal with special tables (large table exclusion), refer to SAP Note 932032.
<i>Number of Entries per Commit</i>	To improve performance, enter a value that is greater than the amount of entries for conversion. If the limit of the database roll area is lower, the tables are not updated in one step. You can determine the required

Field	Description
	value with report RBDLS_CHECK.

3. Choose the required job execution mode to run the conversion:

Execution Mode	Description
Job execution mode 1: <i>Single Job Execution</i>	<p>You can select this execution mode only for single pair conversion. The conversion is done exactly as it is done in transaction BDL5. The tables are processed sequentially using program RBDLSMAP2 in a single background job SAP_SCTC_RBDLSMAP.</p> <p> Example of a created background job</p> <p>SAP_SCTC_BDL5MAP_000 - All tables can be displayed in a spool list.</p>
Job execution mode 2: <i>Concurrent Jobs based on Table Name</i>	<p>The conversion is done in parallel jobs without pre-analysis of the tables. The tables are grouped alphabetically or with special characters predefined as follows:</p> <ul style="list-style-type: none"> A – V: One job for each character /A - /B: One job /C - /I: One job /J - /Z: One job W – Z, 0 – 9, /0 - /9: One job in total <p>To enable this execution mode, the local server must have at least 4 work processes of type batch.</p> <p>This execution mode executes job RBDLS2LS. Each job converts tables whose names start with the prefix SAP_SCTC_BDL5. In addition, the monitoring job SAP_SCTC_BDL5_SNCH starts, that waits until all concurrent jobs are finished. Then, it executes program RBDLSMAP2 to start job SAP_SCTC_RBDLSMAP for the special tables (exceptional objects) defined in exits. Each table set is assigned to a conversion job and can run independently using program RBDLSMAP2. 26 jobs are created and run in parallel to convert the standard tables. All tables in BDLSEXT and BDLSEXZ where you defined an exception are not processed concurrently.</p> <p> Example of created background jobs</p> <p>SAP_SCTC_BDL5000_1001 - All tables starting with letter A (A*) are converted.</p> <p>SAP_SCTC_BDL5000_1026 - All tables starting with letters W to Z and numbers 0 to 9 are converted.</p> <p>SAP_SCTC_BDL5MAP_000 - Special tables are converted with program RBDLSMAP2.</p>
Job execution mode 3: <i>Concurrent Jobs based on Table Entry</i>	<p>Executes concurrent jobs using program RBDLSMAP2.</p> <p>The number of created parallel background jobs is equal to the number of the configured background work process. The distribution of tables among jobs depends on the sorted order of logical system name entry</p>

Execution Mode	Description
	<p>per table, based on the table analysis or the analysis task result. The table with the highest entry count is processed by the first background job. Tables with lesser entries are processed by the last background jobs. The jobs name start with prefix SAP_SCTC_BDLS. In addition we start the job SAP_SCTC_BDLS_SYNCH. This monitoring job waits until all concurrent jobs are finished. Then it executes program RBDLSMAP2 to start job SAP_SCTC_RBDLSMAP for the special tables (exceptional objects). All tables where you defined an exception are not processed concurrently.</p> <p>For more information about the analysis options, see Optimized Conversion with PCA for BDLS.</p>
<i>Release Jobs</i>	<p>To automatically release all scheduled jobs in the configured clients, select the checkbox. If the checkbox is deselected, the jobs will be created in status planned and you have to release the jobs manually in transaction SM37.</p> <p>It is possible to release jobs with the prefix SAP_SCTC_BDLS<Client> in parallel. Ensure to release job SAP_SCTC_BDLS_SYNCH_<Client> at the end.</p>

4. To set local client conversion, no RFC destination is required. To set multiple client conversion, choose *Maintain RFC Destination*.

Client	Description	RFC Destination
000	SAP AG Konzern	NONE
001	SAP AG Konzern	NEWCLNT000

Figure 2-Conversion in local client

Client	Description	RFC Destination
000	SAP AG Konzern	
001	SAP AG Konzern	DEVCLNT001
200	Test Client Import	

Figure 3-One client conversion in productive client

- o For the local client, maintain destination **NONE**.
- o All destinations are checked against the configured client.
- o The remote client execution is only allowed in the local system.
- o The first client with assigned RFC destination is always set for client dependent and independent table conversion.
- o Additional maintained clients are then set only for client dependent table conversion.
- o The conversion takes place in one client after another.

3 Post-Copy Automation - Optimized BDLS Task

3.1 Overview of Analysis Options

If the runtime achieved with the PCA-BDLS task does not suffice your requirements, you can further optimize by reducing the amount of tables for conversion. If you want to reduce amount of tables for conversion, you analyze the system first according to corresponding logical system names – for this, choose the job execution mode *Concurrent Job based on Table Entry* in the BDLS conversion task.

For job execution mode *Concurrent Job based on Table Entry*, you have several options when and how to run a detailed analysis in order to isolate tables affected by the required logical system name change. All these options use the standard ABAP report `RBDLS_CHECK`, which determines the number of logical systems in relevant tables, but embed this feature in different ways.

Report `RBDLS_CHECK` generates a list of tables and table fields that might contain values that are equal to the given list of logical system names. It counts the number of occurrences of these logical system names within these table fields. The results of the analysis get stored in table `BDLSCOUNTALL`, relevant tables and values are automatically passed to the BDLS conversion task `CL_STCT_SC_CONV_BDLS` as input parameters, which will create separate jobs with specific sets of tables, such that each job converts approximately the same number of table entries.

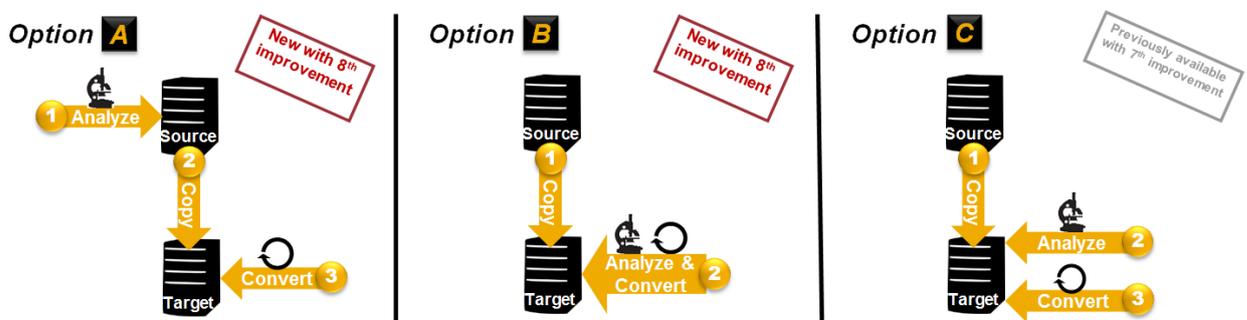


Figure 4 - Analysis Options

- Option A:
You run report `RBDLS_CHECK` for the table analysis before system copy in the source system/original system. The report collects detailed table information for optimizing the BDLS conversion task. This allows a visual inspection of table structure and enables better planning possibilities of your system copy project before the copy procedure. The BDLS conversion task `CL_STCT_SC_CONV_BDLS` can automatically consume the results.
- Option B:
You execute report `RBDLS_CHECK` for table analysis together with execution of the configuration task for BDLS conversion `CL_STCT_SC_CONV_BDLS`. This option optimizes the job distribution. This is interesting especially if it is organizationally not possible to execute analysis prior to task list execution.

- Option C:

You execute a specific configuration task for table analysis prior to executing the BDLS conversion task `CL_STCT_SC_CONV_BDLS`, but after copy.

For this, you use BDLS analysis task `CL_STCT_SC_PREPARE_BDLS` that determines all tables relevant for conversion by executing report `RBDLS_CHECK`. In addition, it determines the optimal number of entries per commit value. The results get stored in table `BDLSCOUNTALL`. Relevant tables and values are automatically passed to BDLS conversion task `CL_STCT_SC_CONV_BDLS` as input parameters. Option C is similar to option B, but allows running the analysis standalone, before the conversion.

3.2 History

With the 7th improvement of system copy task content, only option C of the optimized BDLS execution was offered by PCA. With the 8th improvement of system copy task content, the optimization got further improved with introducing options A and B.

For more information about the 8th improvements, see [SAP Note 2039075](#).

While the previous (7th) improvement for the optimized BDLS scenario must not be used in connection with any SAP Business Warehouse PCA task list (for SAP Business Warehouse and connected SAP Business Warehouse source systems, such as SAP ECC Server or SAP CRM systems), the latest (8th) improvement is also supported for scenarios of SAP Business Warehouse systems and connected SAP applications.



Recommendation

Use the latest version (8th improvement) of the system copy task content for the BDLS improvements.

3.3 Execution

3.3.1 Option A: Execute report for table analysis before system copy in source system

You run report `RBDLS_CHECK` before system copy in the source/original system to determine a list of the corresponding relevant tables which can be used for optimization. An existing analysis from execution of report `RBDLS_CHECK` is used, if the report was executed with corresponding logical system names.

The report `RBDLS_CHECK` can be called using transaction `SE38`, either in the source/original system of the copy or in the local system (the copy), because analysis results are stored persistently and are hence copied together with the system itself.

The execution of the analysis in the original system can reduce the BDLS execution time significantly and is useful, as it allows a visual inspection of the results before the copy procedure. This option enables you to define early your optimization strategy. It improves the planning of your system copy project. This is especially interesting in SAP BW systems, since certain generated SAP BW tables might be excluded from conversion or a change of the data model using SAP Note [1894679](#) might be applied. In any case, the potentially time consuming analysis is then done prior to the time- critical post-copy activities.

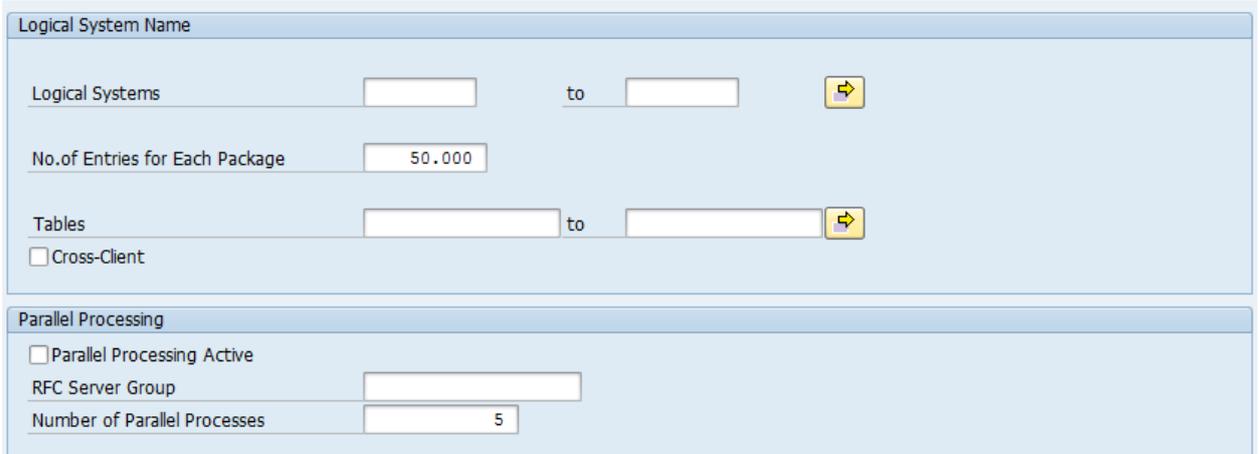


Figure 5 - Report RBDLS_CHECK User Interface

Field	Description
<i>Logical Systems</i>	Enter the logical system names that you want to analyze. To get all information of available logical systems, leave the fields empty.
<i>No. of Entries for Each Package</i>	This field is only relevant for conversion. Enter the number of table entries that are read from the database in each case. To improve performance, enter a value that is greater than the amount of entries for a conversion. If the limit of the database roll area is lower, the tables are not updated in one step. You can determine the required value with report RBDLS_CHECK.
<i>Tables</i>	Specify or exclude tables for analysis, and therefore target specific tables in the database. To determine all related tables, leave the field empty. To determine selected tables, choose  (Multiple selection) and enter the required data.
<i>Cross-Client</i>	To determine the relevant tables for all configured clients, select this checkbox.
<i>Parallel Processing Active</i>	To use additional system resources and speed up processing time, select this checkbox. To determine the table entries for each table in a separate process that is started in parallel, select the checkbox. You can specify a server group and a number of parallel processes that can be handled as a maximum by this report.

If report RBDLS_CHECK is executed before the system copy procedure with the corresponding logical system names, the analysis results are used during the post-copy activities. The BDLS conversion task verifies that the results are valid and, if so, they are used for an optimal parallelization of the conversion. A warning message for the particular logical system name appears when the analysis data is missing or is older than the value set in the

UI (for example 180 days). In addition, any newly created tables are determined that were not part of the analysis. These tables are also logged in a warning message. Confirmation of the warnings triggers an automatic execution of the required analysis by executing report `RBDLS_CHECK` for the necessary selection (missing table or missing logical system). The update of recent changes will not cost as much time as executing the whole table scan of the report again, nevertheless ensure that the analysis result is recent and complete with regards to the intended conversion.

3.3.2 Option B: Run table analysis together with BDLS conversion

You run the table analysis within executing the BDLS conversion.

This option is interesting especially if it is organizationally not possible to execute report `RBDLS_CHECK` prior to the task list execution, meaning that the result remains valid until the execution takes place or the source/original system is not accessible before. Since the analysis and conversion are combined within the BDLS conversion task, only the logical system names entered for conversion will be analyzed.

A new analysis of the logical system names is executed (by calling `RBDLS_CHECK` report) within the BDLS conversion task directly before the conversion to determine the corresponding relevant tables.

Like with the option to use the BDLS analysis task, an overall performance gain can be expected (compared to parallelization by table name), if the histogram is very distinct from an equal distribution, meaning that some tables contain many more entries than most other tables or if the names of the tables cluster for certain letters (e.g. most BW tables start with 'R').

3.3.3 Option C: Execute table analysis prior to executing BDLS conversion

This option allows a visual inspection of the table structure before the system copy procedure itself. The configuration task for BDLS conversion can consume the results automatically.

This option is basically the same as option B, however with the BDLS analysis task `CL_STCT_SC_PREPARE_BDLS` and with the option to run the analysis standalone, before the conversion. This option is only enabled in the BDLS conversion task, if the BDLS analysis task is placed in the same task list.

The user interface of the analysis task is identical with the user interface of report `RBDLS_CHECK`. If you used this option before, you can reuse your corresponding task list variants.

An overall performance gain can be expected (compared to parallelization by table name), if the histogram is very distinct from an equal distribution, meaning that some tables contain many more entries than most other tables or if the names of the tables cluster for certain letters (such as for SAP BW, where most tables start with 'R')

1. Add the following tasks manually to your task list:
 - o `CL_STCT_SC_PREPARE_BDLS`
Optimize BDLS Conversion Task
 - o `CL_STCT_SC_CONFIRM_PREP_BDLS`
Confirmation: BDLS Prepare Task Successfully finished?

i Note

Both tasks can be found with class search pattern *BDLS*

- The tasks must be performed in the following order:

E	C...	S...	L...	Autom. Phase	Component	Task Description	H	P...	Parameter Description
<input checked="" type="checkbox"/>				Prepare	BDLS	Optimize BDLS Conversion Task			
<input checked="" type="checkbox"/>				Prepare	MANUAL_STEP	Confirmation: BDLS Prepare Task Successfully finished?			
<input checked="" type="checkbox"/>				Configuration	BDLS	Conversion of Logical System Names			

This setup enables the transfer of parameters from the preparation task to the conversion task when the UI is accessed. A protocol link will be available after successful running of *Optimize BDLS Conversion Task*, which displays the execution results when the task has successfully finished.

Additionally, parameters turned over to the *Conversion of Logical System Names* task and are checked for consistency during the conversion task check phase. This task is equivalent to execute report RBDLS_CHECK in transaction SE38 manually. If starting the report manually or in a separate task list (without the conversion task), make sure that the results are manually copied to the conversion task and take care that the input parameters and result parameters are consistent.

- Provide the required parameters for task *CL_STCT_SC_PREPARE_BDLS*:

i Note

This task could run a long time but it performs read operations and can be started online.

Logical System Name

Logical Systems: to

No. of Entries for Each Package:

Tables: to

Cross-Client

Parallel Processing

Parallel Processing Active

RFC Server Group:

Number of Parallel Processes:

Figure 6 - CL_STCT_SC_PREPARE_BDLS User Interface

Field	Description
<i>Logical Systems</i>	Enter the logical system names that you want to analyze. To get all information of available logical systems, leave the fields empty.
<i>No. of Entries for Each Package</i>	This field is only relevant for conversion. Enter the number of table entries that are read from

Field	Description
	<p>the database in each case.</p> <p>To improve performance, enter a value that is greater than the amount of entries for a conversion. If the limit of the database roll area is lower, the tables are not updated in one step. You can determine the required value with report RBDLS_CHECK.</p>
<i>Tables</i>	<p>Specify or exclude tables for analysis, and therefore target specific tables in the database.</p> <p>To determine all related tables, leave the field empty.</p> <p>To determine selected tables, choose  (Multiple selection) and enter the required data.</p>
<i>Cross-Client</i>	To determine the relevant tables for all configured clients, select this checkbox.
<i>Parallel Processing Active</i>	<p>To use additional system resources and speed up processing time, select this checkbox.</p> <p>To determine the table entries for each table in a separate process that is started in parallel, select the checkbox. You can specify a server group and a number of parallel processes that can be handled as a maximum by this report.</p>

4. Execute the task list.
5. To display the result, choose *Details* in the task log. It shows the following:
 - Corresponding client
 - Configured logical system name
 - Table name where it is located
 - Field names
 - Corresponding entries in the database row count
6. Confirm that the task *CL_STCT_SC_PREPARE_BDLS* is finished.
Relevant tables resulting from the analysis are automatically passed to task *CL_STCT_SC_CONV_BDLS* as input parameters. The old logical system names and the relevant tables are copied to the corresponding fields.
7. Check the values of task *CL_STCT_SC_CONV_BDLS*:
 1. Check the conversion pairs of the logical system names.
 2. Check the value in the *Number of Entries per Commit* field.
 3. Check if the job execution mode *Concurrent Jobs Based on Table Entry* is selected.
You can only select this checkbox, when the analysis results are available.

Execution Mode	Description
Job execution mode 3: <i>Concurrent Jobs based on Table Entry</i>	Executes concurrent jobs using program RBDLSMAP2. The number of created parallel background jobs is equal to the number of the configured background work process. The distribution of tables among jobs depends on the sorted order of logical system name entry

Execution Mode	Description
	<p>per table, based on the table analysis or the analysis task result. The table with the highest entry count is processed by the first background job. Tables with lesser entries are processed by the last background jobs. The jobs name start with prefix <code>SAP_SCTC_BDLS</code>. In addition we start the job <code>SAP_SCTC_BDLS_SYNCH</code>. This monitoring job waits until all concurrent jobs are finished. Then it executes program <code>RBDLSMAP2</code> to start job <code>SAP_SCTC_RBDLSMAP</code> for the exceptional objects. All tables where you defined an exception are not processed concurrently.</p> <p> Example of created background jobs</p> <p><code>SAP_SCTC_BDLS000_2001</code> - The table with the highest amount of entries is converted.</p> <p><code>SAP_SCTC_BDLS000_2004</code> - The tables with the lowest amount of entries are converted.</p>

4. Check the RFC destination values.
8. Resume the task list.

The consistency check verifies whether the transferred parameters have been changed and with that are no longer identical with the preparation results. Removed parameters are logged as message with a warning. Parameter must not be added. To keep data consistent, adding parameters is logged as message with an error.

4 Best Practices

In case you know your system already very well for example, because you have executed BDLS conversions or system copies before in system and do not face any need to improve conversion speed, choose option *Concurrent Jobs based on Table Name* in the BDLS conversion task.

For more information, see [Post-Copy Automation - BDLS Task](#).

If you observed high BDLS conversion runtimes or have no idea of table distribution in your system yet, choose option *Concurrent Jobs based on Table Entry*. For this, we recommend that you use the advanced procedure:

If you have the possibility to execute report `RBDLS_CHECK` in the original system before system copy, we recommend to do so.

In case your system is an SAP BW system, we recommend that you use following advanced procedure and that you check for big generated SAP BW tables:

- Execute table analysis
To reduce table analysis time in the phase of post-copy activities, we recommend to perform the analysis already in the source system.
- Analyze results of analysis run to define optimization strategy
When performed before system copy, you also get a better estimation of improvable BDLS execution runtime, which improves the planning of your system copy project.
- Execute BDLS conversion task considering the analysis results – resulting in a significantly reduced BDLS execution time

5 Conversion Log Details

The BDLS conversion task log [Details](#) link takes you directly to the background job overview with the corresponding date and time stamps of the corresponding task run. It shows the exact jobs created by the BDLS conversion task, provided that the background jobs are still accessible.

You can display the details of the processed tables as follows:

1. Select the job name and the spool icon.
2. Select the spool list display of the corresponding program name, for example `RBDLSMAP2`.
3. In the output controller, select the ABAP list icon (below the type column).

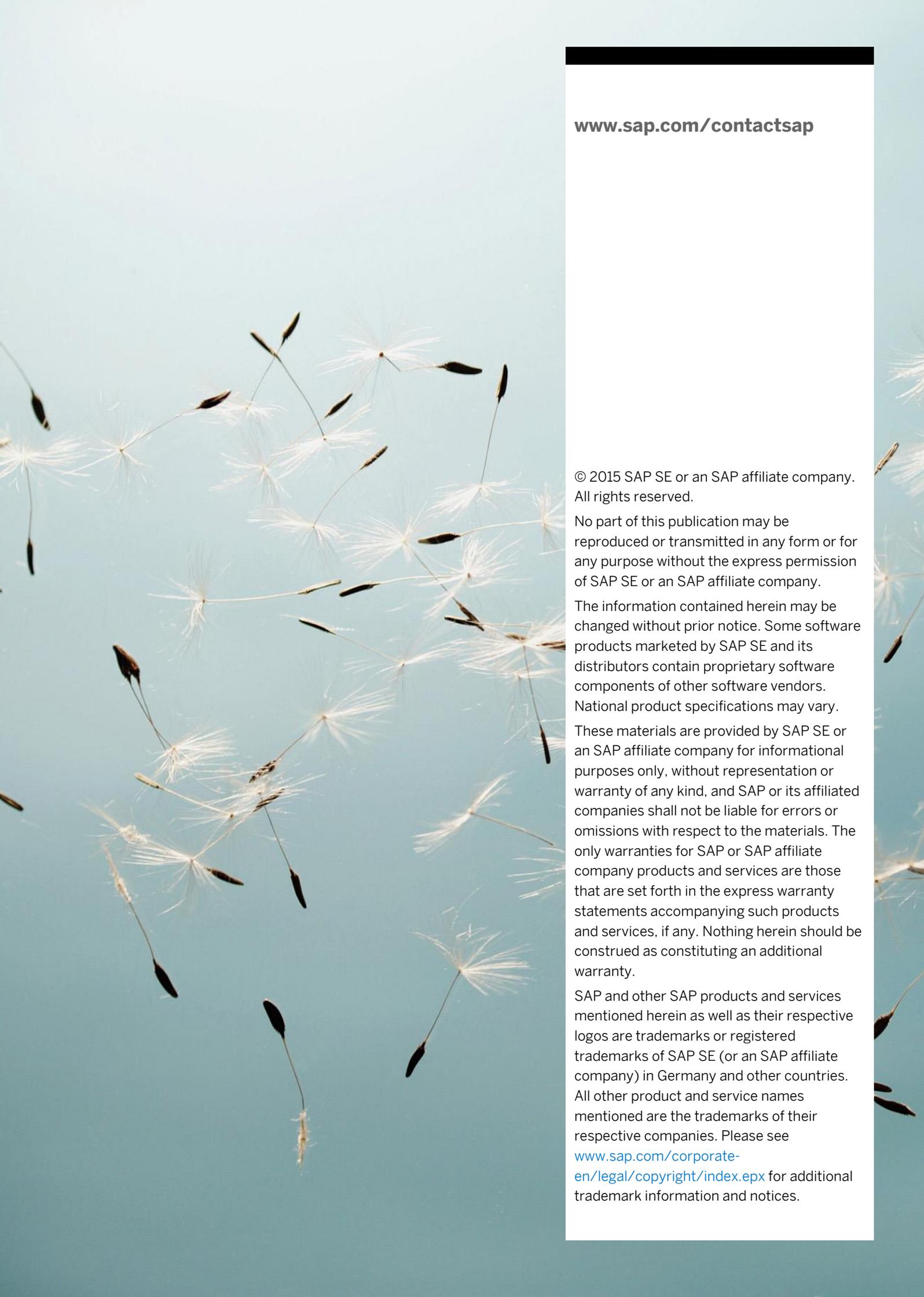
Non-highlighted table names indicate that the table was processed with the corresponding field name, the number of relevant entries, and the number of converted entries. You can view the special conversion only in the last job, for example `SAP_SCTC_BDLSMAP_000`.

To display the exact parameters used for a specific conversion, proceed as follows:

1. Go back to the job overview.
2. Double click the job name, for example `SAP_SCTC_BDLS000_1002`.
3. Select [Step](#).
4. Select the program name.
5. To display the used BDLS program variant, go to [Variant](#).

6 More Information

- For more information about using the post-copy automation content to configure extensive technical scenarios automatically, see the ABAP post-copy automation configuration guide on SAP Service Marketplace at <http://service.sap.com/instrvm>.
- For more information about the BDLS conversion task `STCT_SC_CONV_BDLS` and the BDLS analysis task `STCT_SC_PREPARE_BDLS`, see the description which is available in the *Help* column of the task lists.
- For more information about transaction `BDLS`, search for *Conversion of Logical Systems* within your SAP NetWeaver release on SAP Help Portal at <http://help.sap.com>.
- A central entry point for all issues is the SAP Technology Troubleshooting Guide. For more information, see SAP Community Network at <http://wiki.scn.sap.com/wiki/display/TechTSG>.



www.sap.com/contactsap

© 2015 SAP SE or an SAP affiliate company.
All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. Please see www.sap.com/corporate-legal/copyright/index.epx for additional trademark information and notices.