



New Features Guide

**SAP Replication Server[®] Data
Assurance Option 15.7.1 SP202**

DOCUMENT ID: DC60011-01-1571202-01

LAST REVISED: June 2014

Copyright © 2014 by SAP AG 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 AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group 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 AG in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices.

Contents

New Features in SAP Replication Server Data	
Assurance 15.7.1 SP202	1
Support for Sorting Key Columns in the Index Order	1
Ignore Trailing Spaces in Job Comparisons	2
Redirect Output to a Specific Directory Using the Configuration File	3
Automatic Column Key Assignment for User Defined Compareset Column Mappings	4
New Features in SAP Replication Server Data	
Assurance 15.7.1 SP200	7
Support for All Table Columns as Compareset Key Columns	7
Support for SAP JVM	7
New Features in Replication Server Data Assurance	
Option 15.7.1 SP110	9
Support for IBM DB2 UDB	9
Configuring DA to Use the IBM DB2 UDB JDBC Driver	9
Data Comparison Scenario: Adaptive Server to IBM DB2 UDB	10
Support for Microsoft SQL Server	13
Configuring DA to Use the Microsoft SQL Server JDBC Driver	13
Data Comparison Scenario: Adaptive Server to Microsoft SQL Server	14
Changes to Connection Profile Commands	17
Usability Enhancements	17
Changes to the Run Job Command	17
Changes to the Show History Command	19
Improvements to Large Object Support	19

Changes to Sybase Control Center Data Assurance Plug-In	20
Monitoring and Managing Data Assurance 15.7.1 SP110 Environment	20
New Features in Replication Server Data Assurance	
Option 15.7.1 SP100	21
Support for Sybase IQ	21
Support for Oracle	21
Support for HANA DB	22
Heterogeneous Comparison	22
Data Partitions	23

New Features in SAP Replication Server Data Assurance 15.7.1 SP202

SAP® Replication Server® Data Assurance Option 15.7.1 SP202 supports specifying the order of compareset key columns in **order by** clauses, introduces a comparison option for ignoring trailing spaces in character columns, allows you to direct job output to a directory of your choice, and introduces the **with map option** for compareset commands.

Support for Sorting Key Columns in the Index Order

In SAP Replication Server Data Assurance Option (SAP DA) version 15.7.1 SP202, the **create compareset** and **alter compareset** commands allow you to specify the order of compareset key columns in **order by** clauses using an ordinal integer number in addition to a Boolean value.

Valid values for setting compareset key columns are:

- An integer value ranging from 1 to 2147483647 or 0 to indicate not a key column
- True or false

Note: SAP DA interprets true as 1 or greater, and false as zero (0).

Each key column must be a unique ordinal number within in the compareset. When a compareset has multiple key columns, ordinal integer numbers are used to decide the order of the keys in the **order by** clause. The **order by** clause is included in the select statement, which SAP DA issues to the source and target data servers when selecting data to compare.

For example, when you compare an existing `person` table with these key columns:

```
create table person (
  first_name varchar(50),
  last_name  varchar(50),
  dob       datetime,
  primary key(last_name, first_name)
)
```

The primary keys places the `last_name` column ahead of the `first_name` column, and the data server is optimized for sorting and retrieving rows in this order. A compareset can be created to select the rows in the optimal order:

```
create compareset person
with source conn1 dbo person s
target conn2  dbo person t
map s.first_name = t.first_name set key = 2
and s.last_name  = t.last_name set key = 1
and s.dob = t.dob
go
```

When SAP DA selects rows from the “person” table for a comparison, this SQL query is generated:

```
select first_name,last_name,dob from dbo.person order by
last_name,first_name
```

When you run the **create compareset** command with the **map all** parameter, SAP DA sets the compareset key column ordinal numbers in the optimal order itself.

The pkey column is renamed to key and Y/N values are replaced with numbers in the **show compareset** command output:

```
show compareset person columns
go
```

TYPE	CONNECTION	OWNER	TABLE	MAPID	COLUMN	KEY
S	conn1	dbo	person 1	1	first_name	2
T	conn2	dbo	person 1		first_name	2
S	conn1	dbo	person 2		last_name	1
T	conn2	dbo	person 2		last_name	1
S	conn1	dbo	person 3		dob	
T	conn2	dbo	person 3		dob	

See *Compareset Commands* in the *SAP Replication Server Data Assurance Option Users Guide*.

Ignore Trailing Spaces in Job Comparisons

Use the **ignore_trailing_spaces** new comparison option in the **create job**, **replace job**, and **alter job** commands for ignoring trailing spaces when comparing character datatypes. When the **ignore_trailing_spaces** is true, SAP DA considers “abc” and “abc ” values as equal.

Character datatypes include:

- CHAR
- NCHAR
- VARCHAR
- NVARCHAR
- TEXT
- NTEXT
- CLOB
- NCLOB

The space character is a Unicode value of 'space', U+0020.

Note: Trailing spaces are retained during reconciliation and reporting.

New Parameter:

ignore_trailing_spaces – Determines whether to ignore trailing spaces when comparing character datatype values.

Valid values are:

- true – (default) ignores trailing spaces when comparing character datatype values.
- false – does not ignore trailing spaces when comparing character datatype values.

For example, create a job named “person_job”:

```
create job person_job
add comparison person_cmpson
  set compareset=person_cmpset and
  set ignore_trailing_spaces=true
go
```

See *Row Comparison Job Commands* in the *SAP Replication Server Data Assurance Option Users Guide*.

Redirect Output to a Specific Directory Using the Configuration File

Use the `instance.cfg` configuration file to redirect the output to an alternate directory for storing potentially large reports and reconciliation scripts.

The `instance.cfg` file for SAP DA is found in the default installation directory.

- SAP DA agent:
 - On Windows – %SAP%\DA-15_5\agent\instance
 - On UNIX – \$SAP/DA-15_5/agent/instance
- SAP DA server:
 - Windows – %SAP%\DA-15_5\server\instance
 - UNIX – \$SAP/DA-15_5/server/instance

where *instance* is the name of your SAP DA agent or server.

To redirect the output to a specific directory, open the `instance.cfg` configuration file, and modify the **da.data.dir** parameter to an alternate directory. For example:

```
da.data.dir=${data.instance.dir}/data
```

to

```
da.data.dir=C:/anywhere/server/instance/data
```

Note: The default data directory is authenticated with SAP DA security settings. Change the data directory only when you can secure the data, and potentially large reports and reconciliation scripts are generated.

See *Security and Access Control* in the *SAP Replication Server Data Assurance Option Users Guide*.

Automatic Column Key Assignment for User Defined Compareset Column Mappings

In version 15.7.1 SP202, SAPDA can assign key columns automatically for a compareset with user-defined column mappings.

Use the new **with map option** to map columns in a compareset when there are no key columns.

Note: You can set the **with map option** only for the **strict_key** parameter in the **create compareset** or **replace compareset** command.

When there are no key columns assigned in the compareset, SAP DA searches for any of these:

- Primary key columns. If a primary key is found, SAP DA sets it as a key.
- Identity columns. If an identity column is found, SAP DA sets it as a key.
- Unique index columns. If a uniquely indexed column is found, SAP DA sets it as a key.

If no primary key, identity or unique index columns are found and when the:

- **strict_key** parameter is false (default setting), SAP DA uses all columns as the key.
- **strict_key** parameter is true, SAP DA will not create the compareset and the “no key found” error message is displayed.

Set the **strict_key** parameter using an optional **with map option** as:

```
[with map option set strict_key [{to|=}] {true|false}]
```

For example, if you have a source and target table with these columns:

```
create table person (  
  first_name varchar(50),  
  last_name varchar(50),  
  dob datetime,  
  primary key(last_name, first_name)  
)
```

You can create a compareset without the `set key=true` column option:

```
create compareset person  
with source conn1 dbo person s  
  target conn2 dbo person t  
map s.first_name = t.first_name  
and s.last_name = t.last_name  
and s.dob = t.dob  
with map option set strict_key = true  
go
```

SAP DA will assign the column keys:

New Features in SAP Replication Server Data Assurance 15.7.1 SP202

```
show compareset person columns
go
TYPE CONNECTION OWNER TABLE MAPID COLUMN KEY
-----
S conn1 dbo person 1 first_name 2
T conn2 dbo person 1 first_name 2
S conn1 dbo person 2 last_name 1
T conn2 dbo person 2 last_name 1
S conn1 dbo person 3 dob
T conn2 dbo person 3 dob
```


New Features in SAP Replication Server Data Assurance 15.7.1 SP200

SAP® Replication Server® Data Assurance Option 15.7.1 SP200 includes support for all table columns as compareset key columns, and SAP Java Virtual Machine (JVM).

SAP Replication Server Data Assurance Option 15.7.1 SP200 is compatible with the SAP® Control Center for Data Assurance 3.3 plug-in. For more information, see the SAP Control Center for Data Assurance 3.3 documentation.

Support for All Table Columns as Compareset Key Columns

You can now create comparesets using all non-large object (LOB) columns as compareset keys, when a unique row key column does not exist in the table.

In earlier versions of the SAP Replication Server Data Assurance Option, **create compareset** and **create compareset foreach** commands with **map all** parameter required primary key, identity, or uniquely indexed columns in the source table to create comparesets. In the version 15.7.1 SP200, when a table does not contain a primary key, identity or uniquely indexed column, the **map all** parameter uses all non-LOB columns as the compareset key.

If you do not want to create comparesets with non-LOB columns as the compareset key, set:

- **strict_key** parameter to true in the **create compareset** command
- **column_strict_keys** parameter to true in the **create compareset foreach** command

When all non-LOB columns are used as compareset keys, the keys might not be unique. To abort a comparison when duplicate row keys are detected, use the **abort_dupe_key_immediate** parameter in the **create job** and the **alter job** command.

For command syntax and parameter descriptions, see the *SAP Replication Server Data Assurance Option Users Guide*.

Support for SAP JVM

SAP Replication Server Data Assurance Option 15.7.1 SP200 installer includes the SAP JVM.

For more information about SAP JVM, see the *SAP Help Portal*.

You can modify the DA server `keystore` (flat file) to include the trusted certificate authority (CA).

The list of known and trusted CA certificates are maintained in the `keystore`:

New Features in SAP Replication Server Data Assurance 15.7.1 SP200

- On Windows – `keytool -certreq -alias jetty -keystore %SYBASE%\SCC-3_2\services\EmbeddedWebContainer\keystore -file scc_jetty.csr`
- On UNIX – `keytool -certreq -alias jetty -keystore $SYBASE/SCC-3_2/services/EmbeddedWebContainer/keystore -file scc_jetty.csr`

The **keytool** utility resides in the Java Runtime Environment (JRE) installation directory `%SYBASE_JRE7%\bin\keytool` (on Windows) and `$SYBASE_JRE7/bin/keytool` (on UNIX).

The system administrator can add CA certificates to the `keystore` that exists in SAP Control Center, which is in the `$SCC_HOME/services/EmbeddedWebContainer/cacerts` directory. The default password is `changeit`.

By default, DA server recognizes these third-party CA certificates:

- Baltimore Cyber Trust
- Entrust
- Equifax SCA
- GTE Global
- SAP Passport
- Trustcenter
- Telekom
- Thawte
- VeriSign
- Workplace ca

New Features in Replication Server Data Assurance Option 15.7.1 SP110

Replication Server® Data Assurance Option 15.7.1 SP110 includes support for IBM DB2 Universal Database (UDB) and Microsoft SQL Server as either source or target databases, and usability enhancements.

Replication Server Data Assurance Option 15.7.1 SP110 is compatible with the Sybase® Control Center for Data Assurance 3.2.8 SP01 plug-in. See *Sybase Control Center for Data Assurance 3.2.8 SP01 Documentation*.

Support for IBM DB2 UDB

SAP Replication Server Data Assurance Option 15.7.1 SP110 supports comparisons where the source or target database is IBM DB2 UDB.

You can compare data between any source and target database combinations. For example:

- A homogeneous comparison IBM DB2 UDB to IBM DB2 UDB
- An SAP® Adaptive Server® Enterprise (SAP ASE) source database and an IBM DB2 UDB target database in a heterogeneous comparison
- An Oracle source database with multiple target databases, for example, both SAP Adaptive Server Enterprise and SAP® IQ
- An SAP Adaptive Server Enterprise source database and an SAP HANA® target database

Supported versions:

- IBM DB2 UDB 9.5
- IBM DB2 UDB 9.7
- IBM DB2 UDB 10.1

Data Assurance (DA) does not ship the Java Database Connectivity (JDBC) driver JAR file for IBM DB2 UDB. Download the JDBC type4 driver JAR file from the IBM Web site for your IBM DB2 UDB, and configure DA to use the JDBC driver.

Configuring DA to Use the IBM DB2 UDB JDBC Driver

To use an IBM DB2 UDB database in a comparison, configure the DA server and the DA agent to use the `db2jcc4.jar` file.

1. Download the `db2jcc4.jar` JDBC driver for your database version from the IBM Web site.

2. Copy the `db2jcc4.jar` file into the DA library folder:

- On Windows – `%SYBASE%\DA-15_5\server\lib\`
- On UNIX – `$SYBASE/DA-15_5/server/lib/`

where `%SYBASE%` (Windows) or `$SYBASE` (UNIX) is the directory in which you installed the Data Assurance Option, and `lib` is the library folder of your DA server instance.

3. Restart the DA server for the settings to take effect.

Follow the same steps to configure a DA agent to use the IBM DB2 UDB JDBC driver JAR file.

Data Comparison Scenario: Adaptive Server to IBM DB2 UDB

Perform a heterogeneous comparison using Adaptive Server and IBM DB2 UDB databases.

Prerequisites

Before starting the DA agent, add the JDBC driver JAR file for an IBM DB2 UDB to the DA classpath.

Task

This example uses a DA server local agent connecting to an Adaptive Server and a DA agent connecting to an IBM DB2 Universal database.

Table 1. Deployment Summary

Component Name	Machine Name	Port Numbers
DA server	mars	<ul style="list-style-type: none"> • 6500 – RMI • 6501 – TDS • 6503 – DASD
DA agent	neptune	<ul style="list-style-type: none"> • 6500 – RMI • 6501 – TDS • 6502 – DTS
Adaptive Server database	mars	5000 – server
IBM DB2 Universal Database	neptune	5001– server

1. Start the DA server instance named `mars`:

```
$SYBASE/DA-15_5/server/instance/RUN_instance_64.sh
```

where \$SYBASE is the directory in which you installed the Data Assurance Option, *instance* is the name of your DA server instance, and `RUN_instance_64.sh` is the start-up script.

Note: On Windows, the start-up script file is named `RUN_instance_32.bat` or `RUN_instance_64.bat`. On UNIX or Linux platforms, the file is named `RUN_instance_64.sh`.

2. Start the DA agent instance on the machine named `neptune`:

```
$SYBASE/DA-15_5/agent/instance/RUN_instance_64.sh
```

3. From `isql`, log in to DA server as an administrator:

```
$SYBASE/OCS-15_0/bin/isql -S mars:6501 -U da_admin -P password -w 250
```

4. Create a DA agent connection that connects to the IBM DB2 Universal Database:

```
create agent agent_neptune
  set host=neptune
  and set port=6500
  and set user=da_admin
  and set password=password
go
```

5. View the newly created DA agent:

```
show agent agent_neptune
go
```

6. Test connection settings for the DA agent:

```
test agent agent_neptune
go
```

7. Create database connections for the new DA agent.

In this example, the `pubs2` database is replicated on the IBM DB2 UDB. The DA server's local agent on `mars` points to the Adaptive Server database on `mars`, while the agent on `neptune` points to the IBM DB2 UDB:

```
create connection conn_mars
  set type=ASE
  and set agent=localagent
  and set host=mars
  and set port=5000
  and set user=sa
  and set database=pubs2
  and set password=password
go
```

```
create connection conn_neptune
  set type=UDB
  and set agent=agent_neptune
  and set host=neptune
  and set port=5001
  and set database=PUBS2
  and set user=JOHN
  and set password=mssql12
go
```

8. View the newly created database connections:

```
show connection
go
```

```
show connection conn_mars
go
```

```
show connection conn_neptune
go
```

9. Test the database connections:

```
test connection conn_mars
go
```

```
test connection conn_neptune
go
```

10. Create a compareset to map the entire source table.

This example compares the Adaptive Server and the IBM DB2 UDB data:

```
create compareset authors_demo4
with
  source conn_mars dbo authors s
  target conn_netpune DB2INST1 AUTHORS t
map
  s.au_id=t.AU_ID set key=true
  and s.au_lname=t.AU_LNAME
  and s.au_fname=t.AU_FNAME
  and s.phone=t.PHONE
  and s.address=t.ADDRESS
  and s.city=t.CITY
  and s.state=t.STATE
  and s.country=t.COUNTRY
  and s.postalcode=t.POSTALCODE
go
```

11. View the compareset:

```
show compareset authors_demo4
go
```

12. Create a job.

This job creates a single comparison that uses the compareset defined in step 10. It compares all rows in the authors table:

```
create job authors_job4
  add comparison cmp_authors4
  set COMPARESET=authors_demo4
  and set NUM_PARTITIONS to 1
  and set ENABLE_ROW_COUNT to false
  and set COMPARE_MODE to row_compare
  and set HASH_TYPE to AGENT_HASH
go
```

Note: Set the `HASH_TYPE` comparison option to `agent_hash` for heterogeneous comparison. The `database_hash` comparison option is used only for comparisons between Adaptive Server.

13. Execute the job to compare the data:

```
run job authors_job4
go
```

DA compares equivalent values stored in distinct datatypes accurately.

Support for Microsoft SQL Server

Replication Server Data Assurance Option 15.7.1 SP110 supports comparisons where the source or target database is Microsoft SQL Server.

Supported versions:

- Microsoft SQL Server 2008
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2012

DA does not ship the JDBC driver JAR file for a Microsoft SQL Server database. Download the JDBC type4 driver JAR file from the Microsoft Web site for your database version, and configure DA to use the JDBC driver.

Configuring DA to Use the Microsoft SQL Server JDBC Driver

To use a Microsoft SQL Server database in a comparison, configure the DA server and the DA agent to use the `sqljdbc4.jar` file.

1. Download the `sqljdbc4.jar` JDBC driver for your database version from the Microsoft Web site.
2. Copy the `sqljdbc4.jar` file into the DA library folder:

- On Windows – `%SYBASE%\DA-15_5\server\lib\`
- On UNIX – `$SYBASE/DA-15_5/server/lib/`

where `%SYBASE%` (Windows) or `$SYBASE` (UNIX) is the directory in which you installed the Data Assurance Option, and `lib` is the library folder of your DA server instance.

3. Restart the DA server for the settings to take effect.

Follow the same steps to configure a DA agent to use the Microsoft SQL Server JDBC driver JAR file.

Data Comparison Scenario: Adaptive Server to Microsoft SQL Server

Perform a heterogeneous comparison using Adaptive Server and Microsoft SQL Server databases.

Prerequisites

Before starting the DA agent, add the JDBC driver JAR file for a Microsoft SQL Server to the DA classpath.

Task

This example uses a DA server local agent connecting to an Adaptive Server, and a DA agent connecting to a Microsoft SQL Server.

Table 2. Deployment Summary

Component Name	Machine Name	Port Numbers
DA server	mars	<ul style="list-style-type: none"> • 6500 – RMI • 6501 – TDS • 6503 – DASD
DA agent	pluto	<ul style="list-style-type: none"> • 6500 – RMI • 6501 – TDS • 6502 – DTS
Adaptive Server database	mars	5000 – server
Microsoft SQL Server database	pluto	1433 – server

1. Start the DA server instance named mars:

```
$$SYBASE/DA-15_5/server/instance/RUN_instance_64.sh
```

where `$$SYBASE` is the directory in which you installed the Data Assurance Option, *instance* is the name of your DA server instance, and `RUN_instance_64.sh` is the start-up script.

Note: On Windows, the start-up script file is named `RUN_instance_32.bat` or `RUN_instance_64.bat`. On UNIX or Linux platforms, the file is named `RUN_instance_64.sh`.

2. Start the DA agent instance on the machine named pluto:

```
$$SYBASE/DA-15_5/agent/instance/RUN_instance_64.sh
```

3. From **isql**, log in to DA server as an administrator:

```
$$SYBASE/OCS-15_0/bin/isql -S mars:6501 -U da_admin -P password -w 250
```

4. Create a DA agent connection that connects to the Microsoft SQL Server:

```
create agent agent_pluto
  set host=pluto
  and set port=6500
  and set user=da_admin
  and set password=password
go
```

5. View the newly created DA agent:

```
show agent agent_pluto
go
```

6. Test connection settings for the DA agent:

```
test agent agent_pluto
go
```

7. Create database connections for the new DA agent.

In this example, the pubs2 database is replicated on the Microsoft SQL Server database. The DA server's local agent on mars points to the Adaptive Server database on mars, while the agent on pluto points to the Microsoft SQL Server database:

```
create connection conn_mars
  set type=ASE
  and set agent=localagent
  and set host=mars
  and set port=5000
  and set user=sa
  and set database=pubs2
  and set password=password
go
```

```
create connection conn_pluto
  set type=MSSQL
  and set agent=agent_pluto
  and set host=pluto
  and set port=1433
  and set database=pubs2
  and set user=steve
  and set password=ibmstell
go
```

8. View the newly created database connections:

```
show connection
go
```

```
show connection conn_mars
go
```

```
show connection conn_pluto
go
```

9. Test the database connections:

```
test connection conn_mars
go
```

```
test connection conn_pluto
go
```

10. Create a compareset to map the entire source table.

This example compares the Adaptive Server and the Microsoft SQL Server data:

```
create compareset authors_demo4
with
    source conn_mars dbo authors s
    target conn_pluto dbo authors t
map
    s.au_id=t.au_id set key=true
    and s.au_lname=t.au_lname
    and s.au_fname=t.au_fname
    and s.phone=t.phone
    and s.address=t.address
    and s.city=t.city
    and s.state=t.state
    and s.country=t.country
    and s.postalcode=t.postalcode
go
```

11. View the compareset:

```
show compareset authors_demo4
go
```

12. Create a job.

This job creates a single comparison that uses the compareset defined in step 10. It compares all rows in the authors table:

```
create job authors_job4
    add comparison cmp_authors4
    set COMPARESET=authors_demo4
    and set NUM_PARTITIONS to 1
    and set ENABLE_ROW_COUNT to false
    and set COMPARE_MODE to row_compare
    and set HASH_TYPE to AGENT_HASH
go
```

Note: Set the **HASH_TYPE** comparison option to `agent_hash` for heterogeneous comparison. The `database_hash` comparison option is used only for comparisons between Adaptive Server.

13. Execute the job to compare the data:

```
run job authors_job4
go
```

DA compares equivalent values stored in distinct datatypes accurately.

Changes to Connection Profile Commands

Replication Server Data Assurance Option 15.7.1 SP110 accepts MSSQL and UDB values in the **type** parameter, when you create Microsoft SQL Server and IBM DB2 UDB connections using the **create connection** command.

You can also use the **type** parameter to enter MSSQL and UDB values in the **replace connection** and **test connection config** commands.

As of 15.7.1 SP110, you can specify an optional `sessionTimeZone` property for an Oracle database connection.

This property is used by the Oracle JDBC driver, when comparing `TIMESTAMP WITH LOCAL TIME ZONE` columns. For example:

```
create connection ora_conn1
    set type=ORACLE
    and set agent=agent_jupiter
    and set host=jupiter
    and set port=1521
    and set database=XE
    and set user=harry
    and set password=openSesam3
    with properties
        set sessionTimeZone='Europe/London'
go
```

The `sessionTimeZone` property overrides the default value.

See *Replication Server Data Assurance Option Users Guide*.

Usability Enhancements

Replication Server Data Assurance Option 15.7.1 SP110 includes several changes to improve usability.

Changes to the Run Job Command

To determine whether a job has completed, you can now use the **run job** command with the **wait** parameter rather than issuing multiple **monitor job** commands.

Earlier versions of DA required you to execute separate commands to run a job and monitor the comparison results. For example:

```
run job myJob
go

monitor job myJob [myComparison]
go
```

While waiting for DA server to complete the job, you might have had to execute the **monitor** command multiple times.

Now, you can execute a single **run job** command with the **wait** parameter and view the comparison results immediately when the job completes. For example:

```
run job myJob wait [timeout seconds]
go
```

where:

- **wait** – the **isql** prompt does not return until DA server completes the job.
- **timeout** – (optional) specifies a value, in seconds, after which you can regain control of the **isql** prompt if, for example, the job is taking too long to complete. Valid values are 1 to 2147483647; if you do not specify a **timeout** parameter value, the **isql** prompt waits indefinitely.

Example 1 – waits indefinitely until the job is completed. You cannot regain the **isql** prompt control until DA server completes the job. The job is considered complete when all comparisons run successfully, or if there are errors that abort the job, or if you abort the job manually using a different **isql** prompt:

```
run job myjob wait
go
```

The returned result is:

SUBMIT TIME	FINISH TIME	COMPARISONS	READ (S)	DIFFS	M	O	I	R
2013-06-27 10:20:51	2013-06-27 10:26:59	1	10000	0	0	0	0	0

(0 rows affected)

In this example, you regain control of the **isql** prompt after 6 seconds approximately (the difference between the Submit Time and the Finish Time). The job is completed and the output is identical to that of `show history myjob latest summary` command.

Example 2 – waits for 10 seconds before transferring control of the **isql** prompt to you. The job is considered complete when all comparisons run successfully, or if there are errors that abort the job, or if you abort the job manually using a different **isql** prompt:

```
run job myjob wait 10
go
```

The returned result is:

SUBMIT TIME	FINISH TIME	COMPARISONS	READ (S)	DIFFS	M	O	I	R
2013-06-27 10:31:52	2013-06-27 10:38:02	1	10000	0	0	0	0	0

(0 rows affected)

In this example, you regain control of the **isql** prompt after 6 seconds approximately (the difference between the Submit Time and the Finish Time). The job is completed and the output is identical to that of `show history myjob latest summary` command.

Example 3 – waits for 5 seconds before transferring control of the **isql** prompt to you. The job is considered complete when comparisons run successfully, or if there are errors that abort the job, or if you abort it manually using a different **isql** prompt:

```
run job myjob wait 5
go
```

The returned result is:

```
COMPARISON PART STATUS  SUBMIT TIME          END TIME RUN PROGRESS NEXT
RETRY ERROR
-----
-----
mycmp      0    RUNNING 2013-06-27 10:49:54          1    89%
[#103] Waited 5 seconds, the job is still running.

(0 rows affected)
```

In this example, you regain control of the **isql** prompt after 5 seconds and the output shows the current 'running' state of the job. The job is still running and the output is identical to that of `monitor job myjob` command.

Changes to the Show History Command

As of Replication Server Data Assurance Option 15.7.1 SP110, the **show history** command no longer shows report files to avoid generating potentially very large report files, when you want to view the history.

In earlier versions of DA, when you executed **show history**, report files were generated along with the history in a potentially large file. For example:

```
show history myJob
go
```

Now, you must execute the **show report** command to view the report files. For example:

```
show report myJob 12
```

```
show report myJob latest
```

Improvements to Large Object Support

In Replication Server Data Assurance Option 15.7.1 SP110, large object (LOB) support is improved, with an increase in the maximum value of the **lob_fetch_size** parameter, for comparing and reconciling LOBs of greater length.

Changes to Sybase Control Center Data Assurance Plug-In

Sybase Control Center Data Assurance plug-in now supports Microsoft SQL Server and IBM DB2 UDB for creating database connections.

See *Document Changes > SCC for Data Assurance Help: New Supported Database Platforms in the Sybase Control Center 3.2.8 ESD #2 Release Bulletin*.

Monitoring and Managing Data Assurance 15.7.1 SP110 Environment

Use Sybase Control Center (SCC) 3.2.8 ESD #2 with SCC for Data Assurance 3.2.8 SP01 to monitor Data Assurance (DA) server, and manage DA agent and database connections.

New Features in Replication Server Data Assurance Option 15.7.1 SP100

Replication Server® Data Assurance Option 15.7.1 SP100 includes support for Sybase® IQ, Oracle, and SAP HANA® as either source or target databases, data partitions for large tables, and heterogeneous comparison enhancements.

Replication Server Data Assurance Option 15.7.1 SP100 is compatible with Sybase Control Center 3.2.8. See *Sybase Control Center 3.2.8 for Data Assurance Documentation*.

Support for Sybase IQ

Replication Server Data Assurance Option 15.7.1 SP100 supports comparisons where the source or target database is Sybase IQ.

Supported versions:

- Sybase IQ 15.4
- Sybase IQ 15.3

The jConnect™ for JDBC™ driver included with DA server and DA agent connects to a Sybase IQ database. No further configuration is required.

See *create connection* in the *Replication Server Data Assurance Option Users Guide*.

Support for Oracle

Replication Server Data Assurance Option 15.7.1 SP100 supports comparisons where the source or target database is Oracle.

Supported versions:

- Oracle 11g
- Oracle 10g

Note: Data Assurance (DA) does not ship the Java Database Connectivity (JDBC) driver JAR file for an Oracle database. You must configure DA to use the Oracle JDBC driver, which is deployed with your Oracle installation.

See *create connection* and *Configuring DA to Use the Oracle JDBC Driver* in the *Replication Server Data Assurance Option Users Guide*.

Support for HANA DB

Replication Server Data Assurance Option 15.7.1 SP100 supports comparisons where the source or target database is HANA DB.

Supported version:

- SAP HANA 1.0

Note: DA does not ship the JDBC driver JAR file for the HANA DB. Download the JDBC type4 driver JAR file from the SAP® Web site for your HANA DB version, and configure DA to use the JDBC driver.

See *create connection* and *Configuring DA to Use the HANA DB JDBC Driver* in the *Replication Server Data Assurance Option Users Guide*.

Heterogeneous Comparison

Replication Server Data Assurance Option 15.7.1 SP100 compares row data between two or more databases in a heterogeneous comparison environment.

A heterogeneous environment means two or more of the databases in use are from different vendors.

You can compare data between any source and target database combinations. For example:

- An Adaptive Server source database and a Sybase IQ target database
- An Oracle source database with multiple target databases, for example, both Adaptive Server and Sybase IQ.
- An Adaptive Server source database and an HANA target database

DA accurately compares equivalent values stored in distinct datatypes. For example, the value 1 stored in a Sybase IQ `NUMERIC` column is equivalent to the value 1 stored in an Oracle `NUMBER` column, and to 1.0 stored in an ASE `FLOAT` column.

To use heterogeneous comparison:

- Configure DA to use the JDBC drivers for Oracle and HANA DB instances, if your heterogeneous comparison includes these databases.
- Create connections for each new database.
- Set the **hash_type** comparison option to `agent_hash`.

See *create connection* and *Heterogeneous Comparison Deployment* in the *Replication Server Data Assurance Option Users Guide*.

Data Partitions

Replication Server Data Assurance Option 15.7.1 SP100 allows large tables to be split into logical partitions.

For maximum performance, you can run each logical partition in parallel on a database with multiple engines. The data partitions can be run in smaller groups, if DA is configured to run fewer comparisons at the same time.

Note: When you run a comparison with a new compareset for the first time, DA runs the comparison with a single partition to gather partition information for subsequent runs.

The SQL **where** clause on the compareset key columns defines the upper and lower partition boundaries. The boundary key values are sampled for a compareset when a comparison that uses it is run for the first time. These boundary samples are stored in the Data Assurance System Database (DASD).

On subsequent runs, any comparison that uses the same compareset uses those boundary samples to split a table into as many logical partitions as required. Each time a comparison is run, DA collects the new boundary samples for the compareset in use, and updates them in the DASD.

To view and manage the data partitions and boundary samples, use:

- **show boundary** and **drop boundary** commands to view or delete the data partition boundary samples for each compareset.
- **NUM_PARTITIONS** comparison option to set the number of data partitions for a comparison.
- **PART** column in the **monitor job** and **show job** command's result set to view the data partition number for a comparison.

See *Row Comparison Job Commands* and *Data Partition Commands* in the *Replication Server Data Assurance Users Guide*.

