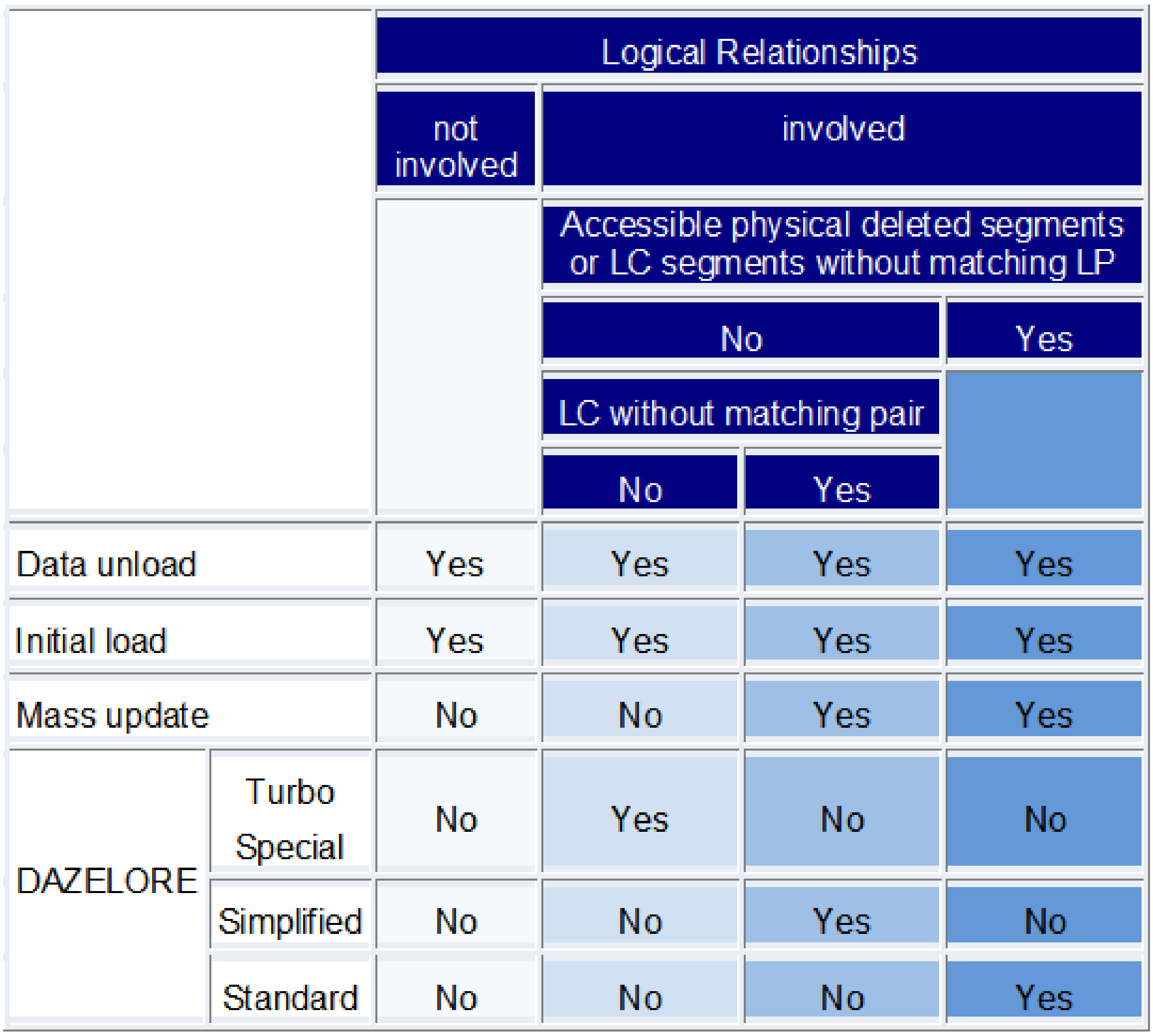
Conversion of the Data - General Considerations

Once the DL/I hierarchical data structures are converted into an Adabas file layout, the data may be moved to Adabas. The steps which are necessary to convert the data depend on the DBDs and user applications involved. The following table lists the necessary steps according to the different prerequisites:



		Logical Relationships			
		not involved	involved		
			Accessible physical deleted segments or LC segments without matching LP		
			No		Yes
			LC without matching pair		
			No	Yes	
Data unload		Yes	Yes	Yes	Yes
Initial load		Yes	Yes	Yes	Yes
Mass update		No	No	Yes	Yes
DAZELORE	Turbo Special	No	Yes	No	No
	Simplified	No	No	Yes	No
	Standard	No	No	No	Yes

This chapter covers the following topics:

- Conversion of the Physical Hierarchical Structures
- Additional Effort Related to Logical Relationships
- Validating Data Types

Conversion of the Physical Hierarchical Structures

The data conversion of a hierarchical structure with no logical relationship involved is quite simple. The data has to be unloaded with the ADL unload utilities and then loaded into an Adabas file, using standard Adabas utilities.

Two different operations are required in order to unload the data from the DL/I database. The first is to read the data and the second is to prepare it for the loading into Adabas.

ADL offers two different methods for performing these steps: an automated and a manual procedure. For the automated procedure, both steps are combined in the utility `DAZUNDLI`, while the manual procedure splits them up into two utilities, namely `DAZUNLOD` and `DAZREFOR`.

It is recommended to use the automated procedure as far as possible. There is no need to customize, assemble or link-edit the program, and no overhead caused by writing and reading the intermediate unload file as with the manual procedure.

Nevertheless, it can be advantageous to choose the manual procedure in some cases. One reason could be that the limited data editing capability of the automated procedure is not sufficient for your applications. Another reason could be that the DL/I system and the ADL nucleus (or Adabas) are not available at the same time. In this case you cannot use the automated procedure, because `DAZUNDLI` accesses both DL/I and ADL. On the DL/I side you have to use `DAZUNLOD`, which runs as a normal DL/I application program. Then move the intermediate unload file to the ADL / Adabas side and run it against `DAZREFOR`, which is a normal mode ADL application program.

Regardless of which method was used for unloading the data, it will be initially loaded into an Adabas file by using the Adabas utilities `ADACMP` and `ADALOD` (`LOAD` function). If no logical relationship is involved, then the data conversion process is finished with the `ADALOD` run.

Additional Effort Related to Logical Relationships

If the database is involved in logical relationships, up to two further steps are required in addition to the unloading and the initial loading of the data. The first is the mass update for paired logical child segments and the second is the establishing of logical relationships.

The mass update is only required if the database is involved in bi-directional logical relationships and if the simplified or standard procedure is used to establish logical relationships. You may use the standard Adabas utilities `ADACMP` and `ADALOD` (`UPDATE` function) in order to perform the mass update.

To establish logical relationships, ADL offers four different procedures: Standard, Simplified, Special and Turbo. All of these procedures use the ADL utility `DAZELORE`. The differences between them are described in detail in the next section.

The Standard procedure can be used in any case, but it is the most time consuming procedure. On the other end, the Turbo procedure has the highest performance but also the most restrictions.

Thus you should use the Turbo procedure whenever possible, or when the Turbo procedure cannot be used, the Special procedure, then the Simplified. Only in cases which cannot be satisfied by any of these three procedures, use the Standard procedure.

Validating Data Types

As described in the section *Conversion of the Data Structures - General Considerations* in this documentation, a DL/I field can contain non-numeric data, even if it is defined as numeric. ADL will convert the DL/I field definition to a numeric Adabas field. During the loading of the data, Adabas will reject the non- numeric values, and thus the load fails.

The standardized ADL unload utility `DAZUNDLI` offers you a limited editing capability for such incorrect data. If you specify the `MODE=CHECKNUM` or the `SEGM/FIELD` parameters, all numeric fields or the specified numeric fields will be checked for valid contents. If ADL detects an incorrect value, a null value (zoned decimal or packed zero depending on the field type) is substituted.

If this limited data editing is not sufficient for you, the customized ADL unload utility `DAZUNLOD` may be used after modifying it as desired.

Nevertheless, to avoid all such failures during the load, you should consider adapting the DL/I field definitions prior to the conversion, as described in the section mentioned above.