Enhancements

This document describes the enhancements that have been made to ADL version 2.3. It is divided into the following sections:

- Changes Related to Adabas
- Changes Related to Natural
- Enhanced Usage of ADL in Batch, CICS and IMS/TP
- Enhanced ADL Conversion and Maintenance tools
- Enhanced ADL Online Services
- Other enhancements

Changes Related to Adabas

This section covers the following topics:

- Support of Adabas 4-byte ISN
- Support of Adabas 2-byte DBID and File Number

Support of Adabas 4-byte ISN

ADL 2.2 supported only 3-byte ISNs which restricted the maximum ISN value to 16.777.215. ADL 2.3 supports 4-byte ISNs which allows having up to 4.294.967.293 records in a file.

Support of Adabas 2-byte DBID and File Number

ADL 2.2 supported only 1-byte DBIDs and file numbers which restricted the maximum value to 255. ADL 2.3 supports 2-byte DBIDs and file numbers which allows having DBIDs in the range of 1 - 65535 and file numbers in the range of 1 - 65534. The ADL directory DBID and file number can be in the range from 1 to 32767.

Changes Related to Natural

This section covers the following topics:

- ADL Online Services Support of New Natural Versions
- ADL Consistency Support of New Natural Versions
- Table of Converted Files
- More Detailed ADL Consistency Status

- De-synchronization of the ADL CALLDLI and Native Adabas
- Natural for DL/I to Natural/Adabas Conversion
- Natural for DL/I INQY (Inquiry) Call

ADL Online Services Support of New Natural Versions

The ADL Online Service is cataloged with Natural 4.1. It is able to run under all Natural versions supporting Natural 4.1 cataloged programs.

ADL Consistency Support of New Natural Versions

The ADL Consistency supports all Natural versions currently supported by Software AG.

Table of Converted Files

The ADL Consistency table of non-converted Adabas files DAZNCF has been replaced by the table of converted Adabas file DAZTCF. DAZTCF supports 2-byte DBIDs and file numbers. For further information see *Consistency Interface* in the section *Batch Installation and Operation* of the *ADL Interfaces* documentation.

More Detailed ADL Consistency Status

The subprogram ADLACTIV, which verifies whether the ADL Consistency is active, returns additional response codes. For further information see the section *Using ADL Files with Natural/Adabas* in the *ADL Interfaces* documentation.

De-synchronization of the ADL CALLDLI and Native Adabas

Adabas calls issued by the ADL CALLDLI Interface and calls issued directly in the program can be de-synchronized with the parameter setting `ADAUSR=YES`. For further information see the section *CICS Installation and Operation* in the *ADL Interfaces* documentation.

Natural for DL/I to Natural/Adabas Conversion

The new section *Converting “Natural for DL/I” Programs* in the *ADL Interfaces* documentation describes what you can achieve and what you must respect when you convert a Natural for DL/I program to a “normal” Natural/Adabas program.

Natural for DL/I INQY (Inquiry) Call

ADL supports the INQY (Inquiry) call issued by Natural for DL/I.

Enhanced Usage of ADL in Batch, CICS and IMS/TP

This section covers the following topics:

- Improved CALLDLI interface

- Language Parameter for ADL Batch Interface
- Simplified Batch TRACE Parameter
- Support of New CICS Versions
- CICS Release Independent Language Interface
- CICS SVC
- CICS Storage Protection
- Storage and Buffer Allocation under CICS
- Storage and Buffer Allocation under IMS/TP

Improved CALLDLI interface

- Call with implied (i.e. missed level) SSA has been revised.
- The delete and replace logic after an unsuccessful path call has been revised.
- Internal fields are only read from Adabas when needed. This reduces not only the amount of data to be shipped but can also improve performance especially when the Adabas Multifetch feature is used.
- The symbolic restart (XRST call) in batch has been revised.

Language Parameter for ADL Batch Interface

A new parameter `LANG` (language) for ADL batch interface DAZIFP is available with the same meaning as the `LANG` parameter in the PSB. It overwrites the language of the PSB. This allows for example to run a DAZZLER stream against a PL/I PSB without the need to generate a new PSB.

Simplified Batch TRACE Parameter

The `TRACE` parameter of DAZIFP can be specified without trailing commas, e.g.

```
TRACE=(RD)
```

Support of New CICS Versions

ADL 2.3 runs under CICS TS 3.2 and below. Note that ADL 2.2 offered only a restricted support for CICS TS 3.1 and did not support CICS TS 3.2.

The ADL load library contains modules which are CICS release dependent. The members in the standard z/OS load library are for CICS TS 3.1 and CICS TS 3.2. When you run under CICS TS 2.3 or below, the ADL.LC23 library must be concatenated in front of the ADL load library.

Under z/VSE, the ADL load library contains the CICS TS 1.1 modules.

CICS Release Independent Language Interface

The ADL 2.2 language interface (DAZLICI2) for z/OS CICS programs was CICS release dependent and did not support CICS TS 3.1 and above. The old language interface is replaced by the new language interface DAZLICI3 which is CICS release independent. All programs which have been linked with DAZLICI2 must be relinked with DAZLICI3.

The ADL language interface provides a new entry point “CEETDLI” to support the IBM language environment (LE) compiler.

CICS SVC

ADL requires an SVC to get control under z/OS CICS. You must specify the SVC parameter in the ADL parameter module and install the ADL SVC in CICS as described in the section *CICS Installation and Operation* of the *ADL Interfaces* documentation.

CICS Storage Protection

CICS applications running against the ADL can use the CICS storage protection feature.

Storage and Buffer Allocation under CICS

ADL 2.3 allocates most storage and buffers under CICS above the line. Some areas must still be allocated below the line because of restrictions when running against HLPI in mixed mode.

ADL has reduced the frequency of storage allocation to improve the performance of online applications.

Storage and Buffer Allocation under IMS/TP

ADL has reduced the frequency of storage allocation to improve the performance of online applications.

Enhanced ADL Conversion and Maintenance tools

This section covers the following topics:

- ADL Conversion Utility Enhancements
- Pseudo mixed mode for unload utility
- Performance improvement for building up logical relationships
- Simplified handling of migrated files

ADL Conversion Utility Enhancements

- The Logical ID of the DBD/segment can be specified at the DBD conversion with the LOGID parameter of the GENDBD/GENSEG function. The logical ID replaces the former DBID and file number info in the ADL internal fields. See *ADL Conversion Utilities for DBDs and PSBs* in the *ADL Conversion* documentation for more information.

- It is possible to omit the DBID parameter at the DBD conversion. At runtime the DBID of the ADL directory will be taken for such a DBD. This easies the creation of mirror databases. For more information see the section *Managing ADL Files* in the *ADL Interfaces* documentation.

- A segment may have the same name as the DBD.

- When loading segments with variable length, the data is correctly initialized.

Pseudo mixed mode for unload utility

The ADL unload utility DAZUNDLI can run in *pseudo mixed mode*. This simplifies the reorganization of migrated files. Pseudo mixed mode is described in detail in the section *Managing ADL Files* of the *ADL Interfaces* documentation.

Performance improvement for building up logical relationships

When DBDs are connected with logical relationships, the ADL utility DAZELORE must run to connect the corresponding data. This must be performed at the initial load and at the reorganization of the data. A new run mode TURBO has been defined for the DAZELORE utility. This mode is much faster than the SPECIAL procedure. With the TURBO version of DAZELORE about 70% of the elapsed time can be saved, e.g. a run which has required 11 hours with the SPECIAL procedure, has only required 3.3 hours with `MODE=TURBO`.

The NUMLC parameter has been replaced by the NUMCP parameter, and the new parameters RESTART and MAXDPISN have been added.

See *Establishing Logical Relationship* in the section *ADL Data Conversion Utilities* of the *ADL Conversion* documentation for a detailed description.

Simplified handling of migrated files

With the introduction of logical IDs, the concept of logical file numbers has become obsolete. The DAZLDT entries can be removed from the ADL parameter source. The new concept simplifies the handling of migrated files as described in the section *Managing ADL Files* of the *ADL Interfaces* documentation.

Enhanced ADL Online Services

This section covers the following topics:

- List applied zap numbers in the ADL Online Services

- Enhanced List Segment Menu

List applied zap numbers in the ADL Online Services

The numbers of the zaps applied to the ADL CICS nucleus can be viewed in the ADL Online Services. See the section *ADL Online Services* in the *ADL Interfaces* documentation for further details.

Enhanced List Segment Menu

The List Segment Menu in the ADL Online Services displays the logical ID specified for the DBD/segment. See the section *ADL Online Services* in the *ADL Interfaces* documentation for further details.

Other enhancements

This section covers the following topics:

- SQL access to migrated data
- Enhanced support of GSAM databases
- ADL Installation Verification Package

SQL access to migrated data

With the Adabas SQL Gateway, the migrated DL/I data can be accessed quickly and easily by any ODBC, JDBC, OLE DB or .NET standard SQL application. IT and business users also can access the data with SQL-based reporting tools, such as Business Objects, Crystal Reports and MS Office. See the section *SQL Access to the Migrated Files* in the *ADL Interfaces* documentation for further details.

Enhanced support of GSAM databases

A GSAM data base uses the Generalized Sequential Access Method (GSAM) under DL/I. ADL 2.2.2 has supported GSAM under z/OS batch and IMS/TP with some restrictions. ADL 2.3 supports GSAM also under z/OS CICS, z/VSE batch and z/VSE CICS.

With ADL 2.3 the ADACMP cards for a GSAM DBD are generated automatically.

For a detailed description see the section *Migration of a GSAM Data Base* in the *ADL Conversion* documentation.

ADL Installation Verification Package

The ADL Installation Verification Package (IVP) provides you with a full DL/I application environment. It can be used to verify the successful installation of the ADL. It helps to gain experience in the ADL concepts and the various ADL tools. If you do not yet have DL/I or Adabas knowledge, you will learn about the most important terms of the both database systems, and how ADL connect the both. The detailed description can be found in the new chapter *ADL Installation Verification Package* in the *ADL Installation* documentation.