

Natural

Database Management System Interfaces

Version 8.3.8

October 2016

This document applies to Natural Version 8.3.8.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1992-2016 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: NATUX-NNATDBMS-838-20161012

Table of Contents

Database Management System Interfaces	v
1 Natural and Entire Access	1
Purpose of Entire Access	2
Configuring Entire Access	2
2 Natural for Tamino	7
Preparing Tamino	8
Configuring Natural for Tamino Access	8
Authorization Aspects	9

Database Management System Interfaces

This documentation provides an overview of the Natural database management system interfaces and a short summary of their functions.

The following topics are covered:

Natural and Entire Access This document describes how to prepare Natural for its use with Software AG's Entire Access under Windows and UNIX.

Natural for Tamino The Natural interface to Tamino enables Natural users to access data in a Tamino database.



Note: See also *Database Access* (in the *Programming Guide*) on how to access data in Adabas.

1 Natural and Entire Access

- Purpose of Entire Access 2
- Configuring Entire Access 2

This document describes how to prepare Natural for its use with Software AG's Entire Access.

Purpose of Entire Access

Via Entire Access, Natural applications can transparently access multiple relational database management systems (RDBMSs), in both client-server and single-platform environments.

Entire Access has been designed to provide a common SQL-compliant application programming interface (API) for both local and remote database access. It represents a client-server solution for Software AG's database systems Adabas SQL Server and Adabas, as well as for various third-party products, for example, ODBC-compliant databases under Windows.

With Entire Access, a Natural user can access data in a relational database. In general, there is no difference between using Natural with Entire Access and using Natural with Adabas. Entire Access supports Natural DML and SQL statements which can both be used from within *one* Natural program accessing *multiple* database systems.

Natural converts Natural native DML and Natural SQL statements within Natural applications into calls to Entire Access, which performs the required data format translations and transfers these requests to the underlying database system via the appropriate database driver.

All operations requiring interaction with the database are performed by Entire Access. It supports local and/or remote databases and contains a series of database drivers, one for each supported RDBMS, so that multiple heterogeneous RDBMSs can be accessed concurrently from within the same Natural application.

For remote access, the transport media for SQL data are provided by Entire Net-Work, which establishes the communication link, or by TCP/IP.

Configuring Entire Access

- [Modifying the Global Configuration File](#)
- [Connecting Databases](#)
- [Making Entire Access Operational](#)
- [Accessing Database Tables - the Natural DDM](#)

Under Windows, Entire Access can be used to access either local ODBC-compliant databases and/or remote databases on Windows and UNIX hosts.



Note: If you want to access remote databases, you also have to install and use the Entire Access server component on the host platform; for further information and prerequisites, see the Installation chapter of the Entire Access documentation.

To be able to use Entire Access, you have to modify your global configuration file, establish a connection to the database(s) to be used, create database tables with the appropriate database tools, and generate a Natural data definition module (DDM) for each database table to be accessed by a Natural program.

Modifying the Global Configuration File

To be able to work with Entire Access, you have to modify the global configuration file (default name: NATCONF.CFG) as follows:

» To modify the global configuration file

- 1 Invoke the Natural Configuration Utility.

The Natural Parameter Setting window appears with the name of the parameter file currently active or edited displayed in the title bar.

- 2 Select the path `Global Configuration File / DBMS Assignments` in the tree view.

A dialog appears in which you have to specify the `DBID` parameter.

For all SQL databases that can be accessed via Entire Access, you always specify the same DBMS type: `SQL`.

Via the `DBMS` parameter, you must then establish the connection to the actual database as described in the following section.



Note: For each database you want to be accessed by Entire Access, you must repeat the specification of the corresponding `DBID` and `DBMS` parameter.

- 3 Save the updated global configuration file.

Connecting Databases

The connection to the database(s) you want to work with must be established only once for each database. Subsequently, you merely have to log on to Natural and the database connection(s) will be established automatically.

Entire Access can support up to 7 different database drivers. Since the database drivers are reentrant, you can establish multiple separate connections to the same driver via a so-called database connect string.

Database Connect String

The database connect string represents the DBMS parameter you or your system administrator must specify with the Natural **Configuration Utility** during the modification of your global configuration file. It consists of the following syntax:

```
dbms:[db-name][@server-number:host-name!driver]
```

Syntax Element Description:

Syntax Element	Description
<i>dbms</i>	<p><i>dbms</i> is mandatory and specifies any supported driver; currently the following drivers are supported under Windows:</p> <p>For ODBC-compliant databases:</p> <p>ODBC for local and remote ODBC-compliant data sources.</p> <p>For remote database access:</p> <p>For remote access, you have to connect the Entire Access network component by specifying NET; see the Entire Access documentation for further information.</p>
<i>db-name</i>	<p>For ODBC-compliant databases:</p> <p><i>db-name</i> is mandatory and represents the name of your data source as it was defined to the ODBC Administrator.</p> <p>Note: The name of the data source is case-sensitive and must not contain any blanks.</p> <p>For remote databases:</p> <p><i>db-name</i> is the name of your database if your database system requires a name; if not, it must be omitted. See the Entire Access documentation for further information.</p>
<i>server-number</i>	<p><i>server-number</i> is required for remote access only; see the Entire Access documentation for further information.</p>
<i>host-name</i>	<p><i>host-name</i> is required for remote access only; see the Entire Access documentation for further information.</p>
<i>driver</i>	<p><i>driver</i> specifies the database driver to be used; see the Entire Access documentation for further information.</p>

Sample ODBC Connect String

To connect to an ODBC-compliant database with a data source called "sample", specify:

ODBC: *sample*

Making Entire Access Operational

Now Entire Access is connected to the database(s) to be accessed. The user ID of the default administrator is `sag`.

➤ To make Entire Access operational

- 1 Start your local ODBC-compliant database(s) in the usual way; for remote databases, refer to the Entire Access documentation.
- 2 Start Natural.

Entire Access can now be used and you can start a Natural session as usual.

Accessing Database Tables - the Natural DDM

To be able to work with Entire Access, you must create database tables or views using the appropriate tools provided by your database system(s).

To be able to access a relational database table or view with a Natural program, you must then generate a Natural data definition module (DDM) for the defined database table or view.

The DDM name must match the name of the corresponding table or view. Thus, you can create only one DDM for each table or view. It is recommended that the DDM name include the name of the table creator (owner) separated by a hyphen. The DBID in the DDM must correspond to the DBID assigned by the Natural Configuration Utility.

Once a DDM has been defined for a relational table or view, it is possible to access the data stored in this table or view by using a Natural program.

Entire Access translates the statements of a Natural program into SQL statements. It automatically provides for the preparation and execution of each statement.

In addition to the Natural native DML statements, Natural provides Natural SQL statements, which are described in the Natural *Statements* documentation; for further information, see also the Entire Access documentation.

For a description of how to create DDMs from relational database tables, refer to the section *Creating DDMs* in the Natural *Editors* documentation.

2 Natural for Tamino

- Preparing Tamino 8
- Configuring Natural for Tamino Access 8
- Authorization Aspects 9

This document describes how to prepare Natural in order to use it with Tamino.

Preparing Tamino

Natural accesses Tamino through HTTP requests. Therefore, Tamino must be set up with an HTTP server component as a connection to the X-Machine. This can be configured during the Tamino installation. See the Tamino *Installation* documentation for more details.

Prerequisites:

See also *Database Access with Natural for Tamino* (in the *Natural Programming Guide*).

Configuring Natural for Tamino Access

Configuring Natural in order to be able to access a Tamino database involves three important steps:

- Defining a new DBID.
- Defining the DBMS type “XML” for this DBID.
- Introducing a new parameter string for this DBID.

To be able to work with Natural for Tamino, you must modify the global configuration file.

➤ To modify the global configuration file

- 1 Invoke the Natural Configuration Utility.
- 2 Edit by selecting the path `Global Configuration File/DMBS Assignments` in the tree view. A dialog appears in which you must specify the **DBID** parameter.
- 3 Enter a new DBID under which you want to access the Tamino database in the **DBID** column. Valid DBIDs for Tamino access are from 1 to 65535.
- 4 Select the type `XML` in the **Type** column.
- 5 Enter a valid parameter string in the **Parameter** column. See the section [Parameter Strings for Tamino Access](#) for more information.
- 6 Save the global configuration file.

Parameter Strings for Tamino Access

The parameter string for a Tamino database consists of the URL of the database server, the name of the database itself and the name of the collection which is to be accessed within this database.

This following is an example of a generic parameter string:

```
http://<hostaddress>/tamino/<dbname>/<collectionname>
```

Parameter Description:

Parameter	Description
<i>hostaddress</i>	The <i>hostaddress</i> is the address of the machine running the web server which communicates with Tamino.
tamino	The keyword <i>tamino</i> is used by the web server as an identifier to open a connection with Tamino.
<i>dbname</i>	The <i>dbname</i> is the name of your existing database on the Tamino database server.
<i>collectionname</i>	<p>The <i>collectionname</i> is the name of the collection within this database to be accessed from your Natural application.</p> <p>All database accesses are based on collection granularity and not on complete databases. If you want to access a different collection within the same database, you will have to add a separate DBID entry for this in the NATCONF.CFG, as described above.</p> <p>For more information on collections, see the <i>DDM Services</i>.</p>

Sample Parameter String

The following is an example of a valid XML database parameter string:

```
http://taminohost.xyz.com/tamino/employeedb/NATDemoData
```

Authorization Aspects

Natural for Tamino supports the following methods of authorization as provided by Tamino:

- Non-authenticated authorization.
- Authorization using web server authentication.
- Authorization using the internal Tamino authentication.

For a detailed description of the authorization methods refer to the Tamino documentation.

If authorization is necessary, Natural for Tamino will ask for the required user and password combination. Alternatively, the user and password data can be provided in environment variables:

- The variables `XML_HTTP_USER` and `XML_HTTP_PASSWORD` are used for web server authentication.
- The variables `XML_DB_USER` and `XML_DB_PASSWORD` are used for the internal Tamino authentication.

Alternatively, the user and password data can be provided in the table **XML Authorization** in the global configuration file. Here a DBID specific setting is possible. Please refer to the *Configuration Utility* documentation.

When using the environment variables for authorization, each connection to a Tamino database will be established with the same pair of user and password. If different users and passwords should be provided for different DBIDs then the usage of the **XML Authorization** table in the global configuration file or the interactive prompting are recommended.