Introduction to Help Topics

Welcome to DataAtlas, the central point of control over information about your organization's data and applications.

DataAtlas enables you to design databases and then populate, create, update, and generate data definitions for those databases. And through its Modeler component, DataAtlas also provides facilities for modeling relational databases.

To learn more about the features and uses of DataAtlas, choose from one of these topics:

Designer Functions
A quick look at what you can do with DataAtlas Designer.

Dictionary Functions
A quick look at what you can do with DataAtlas Dictionary.

Designer Tasks
A guide to how you use the functions in DataAtlas Designer.

Dictionary Tasks
A guide to how you use the functions in DataAtlas Dictionary.

Windows and Notebooks
Reference information for all the Designer and Dictionary notebooks and windows.

To see a list of the database management systems and compilers DataAtlas supports, go to **Supported Products**.

Supported Products

DataAtlas works with objects that are defined by the following database management systems and compilers.

Database management systems:

⇑ DB2 Version 3, DB2 for MVS Version 4, and DB2 for OS/390 Version 5. These high-end database management products are referred to collectively as **DB2/390** on the user interface and in DataAtlas documentation.

⇑ DB2 Universal Database Version 5, referred to as **DB2 UDB** on the user interface and in DataAtlas documentation.

⇑ DB2 Common Server Version 2. This database management system isn't mentioned elsewhere in the help information, but you can assume that all the DB2 UDB help topics apply to DB2 Common Server as well.

⇑ Oracle Version 7.3.

⇑ IMS/ESA Version 6.

Compilers:

⇑ IBM VisualAge for COBOL for OS/2 and Windows Version 2.0.

⇑ IBM VisualAge for COBOL, Professional for OS/2 Version 2.0.

⇑ IBM VisualAge for COBOL 1.2 Refresh.

⇑ IBM PL/I for OS/2 Professional Version 1.2

⇑ IBM PL/I for Windows Version 1.2

DataAtlas requires PL/I compilers to be at the CSD#4 service level or higher.

Designer Functions

When you design a relational database, you begin by designing a relational model.   It shows how tables and columns relate in terms of keys and constraints. DataAtlas Designer enables you to create a   physical design from a relational model: you add the necessary physical information to the tables for the optimum storage of and access to your data.   With DataAtlas Designer, you perform tasks like these:

⇑ Estimate the data load and work load for the application under design

⇑ Define and associate additional resources (such as indexes, table spaces, buffer pools)

⇑ Decide on the physical adjacency of data (such as clustering indexes)

⇑ Decide on the placement of data (such as partitioning)

To get complete assistance, DataAtlas Designer offers design support through a powerful set of rules that compile and exploit DB2 internal knowledge.   For example, you get:

⇑ Information on potential design problems
⇑ Design proposals
⇑ Validation of current designs

On request, proposed design steps are carried out automatically.

See   Designer Tasks for information on using the DataAtlas Designer functions.

Dictionary Functions

DataAtlas Dictionary lets you import (populate) data definitions from existing production environments into the TeamConnection database. You can also create data definitions and store them in the TeamConnection database. When a data definition is in the TeamConnection database, you can use DataAtlas Dictionary to update it to suit your changing requirements. When you want to put a new or updated data definition back into your operating environment, you can use the generate function to create a workstation file containing definition language that conforms to the definition in the TeamConnection database.

DataAtlas Dictionary's principal functions are:

- Populating
- Reconciling
- Creating and deleting
- Updating
- Generating
- Querying

Populating

Populating is the process of importing data definitions into the TeamConnection database. In populating relational database tables, DataAtlas accesses relational catalog tables, enabling it to populate the storage groups, indexes, views, synonyms, aliases, table spaces, and databases the tables need. DataAtlas accesses IMS data definitions from workstation files after they have been downloaded there. Source code comments on the DBDs and PSBs are also populated into the TeamConnection database. DataAtlas accesses COBOL and PL/I data structures from COBOL COPY files and PL/I include files on the workstation.

You can populate in a "trial run" manner; that is, you can ask DataAtlas to process data definitions without storing them. A report is generated that shows how the definitions would have been stored. If the report satisfies your expectations, you can perform an actual populate operation.

See also the related task information on populating data definitions .

Reconciling

When you populate a data definition, you can choose to reconcile its local data components at the same time. If you do, DataAtlas refers to a mapping table and takes one of the following actions for each local data component that is mapped to a shareable data component in the table:

⇑ If the shareable data component is already in the TeamConnection database, DataAtlas assigns to the local data component the attributes of the shareable data component.

⇑ If the shareable data component is not already in the TeamConnection database, DataAtlas creates a shareable data component there and gives it the attributes of the local data component.

See also the related topic "Creating Shareable Data Components without Reconciling."

Creating and Deleting

DataAtlas Dictionary makes it easy to create objects of the following types:

⇑ **For DB2 UDB :** Table, index, database, and view
⇑ **For DB2/390 :** Table, index space, view, storage group, synonym, alias, table space, and database
⇑ **For Oracle:** Table, table space, index, and view
⇑ **For IMS:** DBD, PSB, and PCB
⇑ **For COBOL and PL/I:** Included source definition
⇑ Shareable data element
⇑ Shareable data structure

You can also delete objects from the TeamConnection database. Before deleting an object, DataAtlas Dictionary shows you related objects that might be affected, and lets you delete them as well.

See also the related task information on creating objects and deleting objects .

Updating

With DataAtlas Dictionary, you can easily update an object's characteristics by changing the specifications in its properties notebook. DataAtlas Dictionary checks each notebook field, as well as any related fields, to help prevent you from entering invalid data.

See also the related task information on updating objects .

Generating

DataAtlas Dictionary can generate source statements from data definitions in the TeamConnection database. The generate process is, in a sense, the opposite of the populate process; it exports DBD and PSB macro statements, COBOL and PL/I source code, and SQL statements (SQL CREATE, DROP, ALTER, LABEL ON, and COMMENT ON) to workstation files. If your generated output runs in an OS/390 environment, you must upload it to the host.

See also the related task information on generating data definitions .

Querying

You can use TeamConnection SQL to run queries about the data definitions stored in the TeamConnection database. DataAtlas Dictionary supplies many queries that you can modify, or you can create your own.

TeamConnection SQL, based on an object data model, is slightly different from standard SQL, which is based on a relational data model. For more information on TeamConnection SQL and the supplied queries, see Chapter 8 of *DataAtlas Dictionary User's Guide* .

See also the related task information on querying the TeamConnection database .

Designer Tasks

DataAtlas Designer assists you in designing database objects for DB2/390. DataAtlas Designer functions support the following types of physical design:

⇑ Creating a new database design
⇑ Optimizing an existing database design

DataAtlas Designer offers you two different design modes:

⇑ Direct design:

You directly enter your design data into structured notebooks.

⇑ Design support:

You make use of DataAtlas Designer's design support.

You can get design support at any stage of design of a database object.

Notebooks are available for designing the following DB2/390 database objects:

⇑ Table
⇑ Column
⇑ Index
⇑ Table space
⇑ Database
⇑ Storage group
⇑ View
⇑ Alternate name (alias or synonym)

Starting Database Design from a Physical Design

To start your database design from a physical design, do the following:

1. Open a physical design in one of the following ways:
   - ⇑ From a workfolder:
     - -Open a workfolder that contains a DB2/390 physical design.
     - -Double-click the physical design.
   - ⇑ From a Relational Design notebook:
     - -Open a Relational Design notebook and go to the DB2/390, DB2 UDB, or Oracle page.
     - -Select a physical design.
     - -Select **Open** .

   In the **Physical Design** notebook, you'll see a separate page for each of the object types that can belong to a physical design. On each page you'll find a list of the objects in the physical design that correspond to the page label.
2. Select an object you want to design, and click a push button that identifies the design action you want to take.

Designing Database Objects Using Notebooks

For the physical design of database design objects DataAtlas Designer provides structured notebooks. The notebook sections indicate the design areas of an object.

You have opened either of the following:

⇑ DataAtlas Dictionary workfolder containing DB2/390 objects
⇑ DataAtlas Designer list of a specific type of DB2/390 objects

You start the design with notebooks like this:

1. Double-click the object type you want to design.

   You see the list of objects of this type.
2. Double-click the object you want to design.

   The notebook for this database object opens.
3. Fill in the necessary fields.
4. Click **OK** .

   Your entries are saved and the notebook closes.

Notebooks are available for designing the following DB2/390 database objects:

⇑ Designing a table object
⇑ Designing a column object
⇑ Designing an index object
⇑ Designing a table space object
⇑ Designing a database object
⇑ Designing a storage group object
⇑ Designing a view object
⇑ Creating an alternate name object (DB2 alias or synonym)

Specifying an Object's General Information

To specify the general information of an object, do the following:

1. Open the notebook of an object to the General page.
2. Specify the listed general information items:
   - ⇑ Relational Database Qualifiers:
     - System
     - Creator
     - (Database)

     Where available, select **Search** to get a list of objects that can be entered into the corresponding field. If you select an object from the list, it will appear in the field. If the list is empty, you must first create the object.
   - ⇑ Name
   - ⇑ Variation
   - ⇑ Revision
   - ⇑ Component

     Select **Search** to see a list of components represented in the TeamConnection database. When you select a component name from the list, it is put in the Component field.
   - ⇑ Description
   - ⇑ History

The object is stored in the TeamConnection database with this general information. You can use the relational database qualifiers or the name to retrieve it again from there.

Once you have stored the object in the TeamConnection database, you cannot change the relational database qualifiers. The other specifications can be changed at any time.

Note:

When you select a creator or a database, the System field is filled automatically, since the names for creator and database contain the name of the corresponding system.

When you select a database or a creator, after the other has already been specified, and the system specifications do not match, the last selection is accepted, and the system name is changed accordingly.

Name

This field contains the name of the object.

The value for the name must be a valid DB2 identifier and cannot be an SQL reserved word.

Required?: Yes, for all DB2/390 objects.

Getting Design Support from DataAtlas Designer

DataAtlas Designer offers the following types of design support:

⇑ Information on potential design problems
⇑ Proposals for and automatic execution of design steps
⇑ Validation of designs

You can request design support in these places:

⇑ From an object notebook
⇑ From the lists of objects in a **Physical Design** notebook
⇑ From a workfolder

Design support is divided into two main phases:

⇑ Tailoring design support actions
⇑ Performing design support actions

Requesting Design Support from a Notebook

To request design support from an object's notebook, select a design function from **Designer** on the notebook's menu bar or from the Designs page of the Table, Index, View, Storage Group, or Table Space notebook. A window shows the actions available for the selected design function.

Design support is divided into two main phases:

⇑ Tailoring design support actions
⇑ Performing design support actions

Requesting Design Support from a Physical Design

A list of the objects that belong to a physical design is displayed on the object-type pages of the **Physical Design** notebook. To get design support for listed objects:

1. Select one or more objects for which you want to get design support:

   To select more than one, hold down the Shift key and click the objects.
2. Click the push button that identifies the type of design support you want for the selected objects.

   A window will show the available design actions.

Design support is divided into two main phases:

⇑ Tailoring design support actions
⇑ Performing design support actions

Requesting Design Support from a Workfolder

To request design support from a workfolder:

1. Open the workfolder and select the DB2/390 objects you want to design.

2. Click **Selected** on the menu bar, and then **Designer** .

3. Select a design function from the submenu.

Tailoring Design Support Actions

Each type of design support offers a number of actions, and each action consists of a number of rules. You can determine which rules should be included and specify their parameters where is appropriate.
From the list of objects , you have selected the objects for which you want to get design support. Or you have decided to get design support for an object from its notebook .

You have selected the type of design support you want to get, and a window shows the actions available for the selected design support.

To tailor your set of design support actions, do the following:

1. Select one or more actions.
2. Click **Rules** .

    The notebook of the selected action opens. If you selected several actions, the notebook of the first-selected action is displayed.

    On the Rules page, you see the rules that make up the chosen action. By default, all the rules are selected.
3. Deselect the rules you do not want to include in the action.
4. Open the notebook page of a rule.

    You see the detailed description of the rule.
5. Where appropriate, change the necessary parameters.
6. Click **OK** .

    Your settings for the selected action are saved, and you see the list of actions again.
7. Repeat these steps, if you selected several actions.

The action settings are valid for one session. If you start a new DataAtlas session, your settings are reset to DataAtlas Designer's default values.

Note: If you have selected several objects, your settings apply to all the objects.

After you have specified the settings, you can let DataAtlas Designer perform the selected actions .

Performing Design Support Actions

From the list of objects , you selected the objects for which you want to get design support. Or you decided to get design support for an object from its notebook . You selected the type of design support you want to get, and a window shows the actions available for the selected design support.

To perform design support actions, do the following:

1. Select the action you want to have performed.
2. Click **Get report** .

   DataAtlas Designer performs the selected actions and produces a report.

Note: For a design proposal, you can request the report for only one action at a time.
For design information and validation, you can request the report for several actions.

Next, you can evaluate the report .

Before you let DataAtlas Designer perform the actions, you can tailor the set of rules and set the necessary parameters.

Evaluating Design Support Reports

This report shows the results of the performed actions for the selected type of design support.

Depending on the type of design support, a report contains information messages, design proposals, or warning messages. The report messages are headed by an icon. With design proposals there is an additional column with the title **Accepted** . It indicates whether a proposal has been accepted or rejected.

A report with information on potential design problems contains information messages. A design validation report contains warning messages. A report with design proposals contains design proposals as well as information messages. Here the information messages inform you about data that are missing but are necessary for the creation of an appropriate proposal.

For all types of messages, you can ask for the action item that caused the message:

1. Click a report item with the right mouse button.

   A pop-up menu opens.
2. Click **Rule** .

   You see the detailed description of the rule on which the action item is based.

Also, for each message, you can look at the affected objects:

1. Click a report item with the right mouse button.

   A pop-up menu opens.
2. Click **Details** .

   A window lists the report item again, and presents icons for the affected objects.
3. Double-click the icon to open the object's notebook and look at its specifications.

With a validation report item or an information report item, you only see an icon for the object you are currently designing, because it's the only one affected. With a design proposal item, you also get icons for the proposed objects. The notebooks of the proposed objects already contain the proposals. You can directly accept the proposals here.

You can accept design proposals from the design item window or from the design proposal report.

Sometimes you see report items where you have to make a choice from several proposals .

From a design proposal report, you can tell DataAtlas Designer to directly execute the proposed items .

Accepting Design Proposals

You have several possibilities for accepting design proposals:

⇑ From the Report Item window
⇑ From the design proposal report

To accept design proposals from the Report Item window, do the following:

1. From the Proposal Report window, select a report item with the right mouse button.

   A pop-up menu opens.
2. Click **Details** .

   A window lists the report item again, and presents icons for the current and for the proposed objects.
3. Double-click the icon of the proposed object or the object with the proposed changes that you want to accept.

   The object's notebook opens. The notebook already contains the proposed values.
4. Click **OK** to leave the notebook.

   The proposed values are saved and you are back in the Report Item window.
5. Close the Report Item window.

   In the report, the corresponding report item is marked **Accepted - Yes** , and the icon gets a check mark.

To accept design proposals from the design proposals report, do the following:

1. Select a proposed item.
2. From the Selected pull-down menu:
   a.Select **Accept** if you agree with the proposal.

      The accepted proposals are marked **Accepted - Yes** , and the icon gets a check mark.
   b.Select **Reject** if you do not agree with the proposal.

      The rejected proposals are marked **Accepted - No** .

Selecting Proposals

Sometimes you have to choose between various possible proposals:

1. Click the multiple proposal report item with the right mouse button.

   A pull-down menu appears.
2. Click **More.**

   A window lists the report item again, and presents the icons of the objects to choose from.
3. Double-click the appropriate object.

   The corresponding notebook opens.
4. Click **OK** to accept and save the proposed values.

   The notebook closes, and you are back in the report item window.
5. Select **Cancel** push button to leave the report item window.

   You see the report again. The report item is marked **Accepted - Yes** , and the icon gets a check mark.

From the design proposal report, you can tell DataAtlas Designer to directly execute the proposed items .

Executing Design Proposals

If you want to save the proposed changes to objects or save the proposed new objects, do the following:

1. Accept the items you want to save by marking them with **Accepted - Yes** .
2. Select **Execute** .

   The proposals that are marked **Accepted - Yes** are executed.

   After execution, the proposals are marked **Executed.** If new objects are created, a window will ask you whether you want the objects to be represented in a workfolder.

Designing a Table Object

The physical design is carried out with notebooks. The notebook sections indicate the different design areas.

You perform the design by filling in the necessary fields on the notebook pages. In addition, DataAtlas Designer offers design support. For the table object, all types of design support are available:

⇑ Information on potential design problems
⇑ Design proposals
⇑ Design validations

The design of the table object consists of the following tasks:

⇑ Specifying general information
⇑ Viewing the shareable table definition
⇑ Viewing the columns
⇑ Adding and deleting columns
⇑ Optimizing the table layout
⇑ Assigning the table to a physical design
⇑ Specifying routines and options
⇑ Creating and modifying a primary key
⇑ Creating and modifying unique keys
⇑ Creating and modifying foreign keys
⇑ Creating indexes
⇑ Assigning the table to a table space and a database
⇑ Creating table partitions
⇑ Modifying table partitions
⇑ Specifying the data load
⇑ Specifying the work load
⇑ Extracting DB2 actual values

Getting Design Support for a Table

You can get design support from anywhere in the notebook.

For the table object, the following design support is available :

⇑ Inform - Information on potential design problems:
- -Identification of tables with default values
- -Identification of tables with potential column problems
- -Identification of tables with potential primary key problems
- -Identification of tables with missing design information
- -Identification of tables without an explicit clustered index
- -Identification of large tables
- -Identification of columns with missing design information
- -Calculation of wasted space

⇑ Propose - Design proposals:
- -Assignment of the table to a table space
- -Creation of a primary key
- -Creation of foreign keys
- -Creation of indexes
- -Creation of a partitioned index for a partitioned table

⇑ Validate - Design validations:
- -Identification of tables with invalid primary or foreign keys
- -Identification of tables with column inconsistencies
- -Identification of incomplete tables

To get design support in any of these design areas, select the appropriate type of design support from the **Designer** pull-down of the Table notebook, and then select the desired action.

Viewing the Shareable Table Definition

To view the shareable table definition, do the following:

1. Open the Table notebook to the Definition page.
2. Click **Open** to open the notebook of the shareable table definition.

The column information related to the table definition is listed in the **Columns** table.

Note:

When you use a shareable table definition, you must be careful: When you change the column definitions, the changes are propagated to all of the table objects using this shareable table definition.

Viewing the Columns

To view all of the columns of a table, do the following:

1. Open the Table notebook to the Definition page.

   You see the columns defined for the table listed in the **Columns** table **.**
2. Click **Open** to open a column's notebook and look at its characteristics.

Adding and Deleting Columns

To add a new column to the table:

1. Open the Table notebook to the Definition page.
2. Click **Create** from the **Columns** table.

   An empty column notebook is displayed.
3. If you want to use shareable data elements , click **Search** on the Definition page of the Column notebook.
4. To disassociate a shareable data element from a column, click **Clear** to delete the data element's name from the entry field. However, the data type values are kept for further use. They can be changed while designing the column .

The specified information is listed in the **Columns** table.

To delete a column from the table:

1. Open the Table notebook to the Definition page.
2. Select the appropriate column from the **Columns** table.
3. Click **Delete** .

The column is deleted from the table object, and its information is removed from the **Columns** table.

Note: Changes in the column information of a table cause a change in the shareable table definition too.

Optimizing the Table Layout

To optimize the table layout, do the following:

1. Open the Table notebook to the Definition page.

    The table's columns and their characteristics are listed in the **Columns** table.
2. Select the column you want to reposition.
3. Click **Move up** .

    The column is moved up in the table.
4. Click **Move down** .

    The column is moved down in the table.

Assigning the Table to a Physical Design

Only, when an object is assigned to a physical design, you can fully exploit the design support functions available for the entire physical design. For example, from the physical design, you can request design proposals for several objects at a time.

To assign the table object to a physical design:

1. Open the Table notebook to the Designs page.
2. Click **Search** to search for an appropriate physical design among the list of existing physical designs.

Specifying Routines and Other Options

To specify the routines and set the options for the table object, do the following:

1. Open the Table notebook to the Options page.
2. Specify the following optional information in the corresponding fields:
    - ⇑ Procedures
        - -EditProc
        - -ValidProc
    - ⇑ OBID
    - ⇑ Audit
        - -None
        - -Changes
        - -All
    - ⇑ Data capture
        - -None
        - -Changes
    - ⇑ Label
    - ⇑ Comment

Creating and Modifying a Primary Key

The primary key clearly identifies the rows of a table. It can consist of one or more columns. Only unique values are allowed in primary key columns.

For the creation of a primary key, you have the choice between two different design modes:

⇑ Directly performing this design step in the notebook
⇑ Getting a design proposal from DataAtlas Designer

If you want to directly create the table's primary key:

1. Open the Table notebook to the Primary Key page.
2. From the **Available** list box, select the columns of which the primary key should be composed.
3. Click **Add** .

The selected columns are listed in the **Selected** list box. The primary key is composed of these columns.

Or, to request a design proposal for the creation of a primary key, select **Propose** from the **Designer** pull-down menu, and then **Create primary key** .

Note:

The primary key is a unique key. If you assign unique keys, you must also assign unique indexes. Otherwise you cannot use the implemented table.

When you want to modify the primary key, do the following:

1. From the **Selected** list box, select the columns you want to remove from the primary key.
2. Click **Remove.**

The selected columns are not part of the primary key any more.

Note:

You cannot modify a primary key that is referenced by a foreign key.

Creating and Modifying Unique Keys

Like primary keys, unique keys are a way of allowing only unique values in a set of a columns.

To create unique keys:

1. Open the Table notebook to the Unique Keys page.
2. Click &lt.NEW> in the **Unique keys** list.
3. From the **Available** list box, select the columns of which the unique key should be composed.
4. Click **Add** .

   The selected columns are listed in the **Selected** list box.
5. Click **Remove** to remove selected columns from the **Selected** list box.
6. Click **Create** .

   The columns of the **Selected** list box are listed as unique key in the **Unique keys** list box, separated by an exclamation mark (!).

   Each line represents one unique key.

If you want to modify a unique key, do the following:

1. Select a unique key from the **Unique keys** list box.

   The corresponding unique key columns are listed in the **Selected** list.
2. Modify the set of unique key columns by adding or removing columns.
3. Click **Modify** .

   The changed set of columns is listed as unique key in the **Unique keys** list.
4. Click **Delete** if you want to delete a selected unique key from the **Unique keys** list.

Note:

If you assign unique keys, you must also assign unique indexes. Otherwise, you cannot use the implemented table.

Creating and Modifying Foreign Keys

The foreign key is used to specify a referential constraint on the table. You can create more than one foreign key on the table.

Parent table columns and foreign key columns of the current, dependent table must match in the following aspects:

⇑ Number of columns
⇑ Data types
⇑ Field procedures

You can create a foreign keys by two different means:

⇑ Do it directly in the Table notebook.
⇑ Execute a design proposal from DataAtlas Designer.

If you want to create foreign keys directly:

1. Open the Table notebook to the Foreign Keys page.
2. Click &lt.NEW> in the **Constraint list** .
3. Select **Search** to search for the parent table whose primary key corresponds to the foreign key.

    The primary key columns of the referenced parent table are listed as **Parent key columns** .

    The columns of the current table whose data type values match the ones of the referenced key columns are listed as **Matching columns** .
4. Select the matching columns that should compose the foreign key.
5. Specify a name in the **Constraint name** entry field.
6. Specify the type of delete rule for the newly specified key.
7. Click **Create** .

    The newly specified foreign key is listed in the **Constraint list** .
8. Repeat these steps if you want to create further foreign keys.

If you want to modify a foreign key, do the following:

1. Select a foreign key from the **Constraint list** .

    All the columns of the table are listed in the **Matching columns** list box. The current columns of the selected foreign key are marked.
2. Modify the set of foreign key columns by marking different columns.
3. Change the constraint name if appropriate.
4. Click **Modify** .

    The constraint with its changed set of columns is listed in **Constraint list** .

If you want to request a design proposal for the creation of foreign keys, select **Propose** from the **Designer** pull-down menu, and then **Create foreign keys** .

Creating Indexes

Create indexes to optimize the access path.

For the creation of indexes, you have the choice between two different design modes:

⇑ Directly performing this design step in the notebook
⇑ Getting a design proposal from DataAtlas Designer

If you want to create indexes directly:

1. Open the Table notebook to the Indexes page.

   In the **Indexes** list, you see all the table's current indexes.
2. Click **Create.**

   An empty index notebook opens into which you can enter the required data. After closing the index notebook, the new index is listed in the **Indexes** list.

To change an index:

1. Select an index from the **Indexes** list box.

   The columns belonging to the selected index are displayed in the **Columns of selected index** list box.
2. Click **Open** to open the corresponding notebook and change the necessary data.

If you want to request a design proposal for the creation of indexes, select **Propose** from the **Designer** pull-down menu, and then select **Create index** .

Assigning the Table to a Table Space and a Database

A table must be stored in an appropriate table space. Assign the table explicitly to a table space to avoid that the DB2 default assignments are used.

To assign a table to a table space, you can choose between two different design modes:

⇑ Directly performing this design step in the notebook
⇑ Getting a design proposal from DataAtlas Designer

To directly assign a table to a table space:

1. Open the Table notebook to the Table Space page.
2. Click **Search** to search for an appropriate table space to which you can assign the table.

The selected table space appears in the corresponding entry field. If the table space is assigned to a database, this database is automatically assigned to the table too, and the name of the database is listed in the corresponding entry field.

If the selected table space is not assigned to a database, no database is directly assigned to the table. DB2 then assigns the table to the default database DSNDB04.

If you do not want to explicitly assign the table to a table space, you can also first select the database. If the database contains exactly one table space, this table space is listed in the table space entry field. If the database contains more than one table space, the table space entry field remains empty. If you do not explicitly select one of the assigned table spaces in the course of your physical design, the DB2 default table space is used then. The DB2 default table space is also used, if the database has no table space assigned.

If you want to request a design proposal for the table to table space assignment, select **Propose** from the **Designer** pull-down menu, and then **Assign table to table space** .

Creating Table Partitions

To optimize the performance, large tables are best assigned to partitioned table spaces. To support the table's assignment to a partitioned table space, DataAtlas Designer lets you create table partitions.

You can specify a partitioned table space or a partitioned index only if you created table partitions.

DataAtlas Designer also proposes a partitioned table space or a partitioned index for a table only if you have created table partitions.

Note:

Table, table space, and index partitions should always match in number.

You create table partitions like this:

1. Open the Table notebook to the Partitions page.
2. Specify the number of partitions you want to create.
3. Click **Accept.**

   The numbered partitions are listed in the **Partitions** table.
4. From the **Available** list box, select the columns for the partition.
5. Click **Add** .

   The selected columns are listed in the **Selected** list box. In addition, they are inserted into the **Partitions** table.
6. Select a partition from the **Partitions** table.
7. Click **Open** to specify range values for the columns of the partition.

   The Table Partitions window opens. The number of the partition is listed. The names of the columns for which you want to create ranges are inserted automatically.
8. For each column, specify the target UPPER value delimiting its range.

   The concatenation of specified column ranges determines the partition.
9. Close the Table Partitions window by clicking **OK** .

   The partition and its specified column ranges are listed in the **Partitions** table.

You can modify a partition at any time.

Note: Be aware that DB2 uses only the first 40 bytes of the concatenated range values. The bytes exceeding this range are ignored.

Modifying Table Partitions

Changes you make to the table partitions do not affect DDL generation. If you want to keep the table partitions as a source for the table space and index partitions, you must either change the partitioned table space and partitioned index manually or request a design proposal for the creation of a new table space and index.

To modify a table partition:

1. Open the Table notebook to the Partitions page.
2. From the **Partitions** table, select the partition you want to modify.
3. Click **Open** .

   The Table Partitions window opens. You see the number of the partition and the current target UPPER values delimiting the ranges of the columns.
4. Change the range values where necessary.
5. Click **OK** to close the Table Partitions window.

   The changes are listed in the **Partitions** table.

Specifying the Data Load

Together with the work load values, the data load values determine the adequate configuration of a table's access and storage structures.

Many of the design support functions DataAtlas Designer offers make use of the data load information. The most important value here is the value for the *initial number of rows.*

To specify the data load values of the table object:

1. Open the Table notebook to the Data Load page.

   You see a read-only table with the data load for the listed partitions.
2. Select a partition.
3. Click **Change** .

   The Table Partitions Data Load window opens.
4. Enter the necessary data load information.
5. Click **OK** to close the window.

   Your values are accepted and listed in the data load table.

Specifying the Work Load

Together with the data load values, the work load values determine the adequate configuration of a table's access and storage structures. DataAtlas Designer requires work load values, for example, for the calculation of index proposals.

The work load information for a table consists in specifications of frequency and concurrency values for a number of operations.

To specify the work load values of the table object:

1. Open the Table notebook to the Work Load page.

   You see a read-only table with the work load for the listed partitions.
2. Select a partition.
3. Click **Change** .

   The Table Partitions Work Load window opens.
4. Specify the necessary frequency and concurrency data.
5. Click **OK** to close the window.

   Your values are accepted and listed in the work load table.

Extracting DB2 Actual Values

Here you extract the most important DB2 statistic values of the DB2 catalog. Together with the statistic values of the table, the statistic values of all the table's columns and assigned indexes are retrieved.

To extract the statistic values of the objects from the DB2 catalog:

1. Open the Table notebook to the DB2 Actuals page.
2. Click **Search** to search for the database that contains the DB2 catalog from which you want to extract the statistic values.
3. Click **Extract statistics** to extract the current DB2 values.

   The values are listed in the corresponding fields.

Note:

No data is available for extraction as long as the table object has not been implemented.

Designing a Column Object

The physical design of the column object is carried out with the Column notebook. The notebook sections indicate the different design areas.

The Column notebook is accessed from the Definition page of the Table notebook.

DataAtlas Designer offers design support. For the column object, you can get design support in terms of Information on potential design problems.

The design of the column object consists of the following tasks:

- Defining a column
- Setting options
- Specifying the data load
- Specifying the work load
- Viewing DB2 actual values

Getting Design Support for a Column

You can get design support from anywhere in the notebook.

For the column object, the following design support is available:

⇑ Inform - Information on potential design problems:
  -Identification of columns with data type problems
  -Identification of columns with missing design information

No Propose or Validate functions are available for the column object.

To get design support in any of these design areas, select the appropriate type of design support from the **Designer** pull-down of the Column notebook, and then select the desired action.

Note: Since columns are part of a table, their design support actions are also part of the ones for the corresponding table.

Defining a Column

To specify the parameters by which the column is defined:

1. Do one of the following on the Definition page of the Table notebook:
   - ⇑ Click **Create** if you want to create a new column.

     An empty Column notebook displays.
   - ⇑ Select a column and click **Open** if you want to work with an existing column.

     The corresponding notebook opens.

2. Open the Definition page of the Column notebook.
3. Specify the listed parameters:
   - ⇑ Name

     This field is mandatory.

     You can change the name of a column at any time.
   - ⇑ Type
     - Precision
     - Scale
     - Length
     - Byte Count

   - ⇑ Subtype:
     -Bit
     -Mixed
     -SBCS
     -None

   - ⇑ Shareable Data Element

     If you want to use predefined shareable data elements, click **Search** to search for the appropriate shareable data element.

     Click **Open** to open the notebook of the shareable data element.

     Click **Remove** to clear the entry field and to disassociate a shareable data element from a column.

A shareable data element has its own data definition. Therefore, if the column is based on a shareable data element, the data definition values are filled in, and the fields are read-only. They can be changed only in the notebook of the shareable data element.

Setting Options

To specify the options for a column:

1. Open the Column notebook to the Options page.
2. Specify the following:
   - ⇑ Field Procedure:
     - -Name
     - -Parameters
   - ⇑ Whether no nulls are allowed
   - ⇑ Whether the column will have a default value and, if so, what it will be

The **Comment** field gives you room for your own notes and comments on the object you are currently designing.

After the table has been implemented, the field procedure cannot be changed, added, or deleted. The field procedure is optional. If you omit it, the column has no field procedure.

Specifying the Data Load

Together with the work load values, the data load values of the column determine the adequate configuration of the storage and access structures of the corresponding table. DataAtlas Designer requires the data load values, for example, for the calculation of index proposals.

To specify the data load values of the column:

1. Open the Column notebook to the Data Load page.
2. Enter the data load information into the listed fields.

Specifying the Work Load

Together with the data load values, the work load values of the column determine the adequate configuration of the storage and access structures of the corresponding table. DataAtlas Designer requires the work load values, for example, for the calculation of index proposals.

To specify the work load values of the column:

1. Open the Column notebook to the Work Load page.
2. Enter the work load information into the listed fields.

Viewing DB2 Actual Values

Statistical values of a column can be extracted through the Extract statistics function on the DB2 Actuals page of the Table notebook.

To view these values:

⇑ Open the Column notebook to the DB2 Actuals page.

You see the most important column statistics from the DB2 catalog.

Designing an Index Object

The physical design is carried out with notebooks. The notebook sections indicate the different design areas.

You perform the design by filling in the necessary fields on the notebook pages. In addition, DataAtlas Designer offers design support. For the index object, all types of design support are available:

⇑ Information on potential design problems
⇑ Design proposals
⇑ Design validations

The design of the index object consists of the following tasks:

⇑ Specifying general information
⇑ Assigning the index to a physical design
⇑ Defining the index
⇑ Specifying index columns
⇑ Specifying the sorting order
⇑ Specifying the storage information
⇑ Viewing index partitions
⇑ Specifying type and storage information of an index partition
⇑ Viewing DB2 actual values

Getting Design Support for an Index

You can get design support from anywhere in the notebook.

For the index object, the following design support is available :

⇑ Inform - Information on potential design problems:
  -Identification of indexes that can result in increased maintenance cost
  -Identification of indexes that could be dropped

⇑ Propose - Design proposals:
  -Calculation of space requirements
  -Setting of options

⇑ Validate - Design validations:
  -Identification of indexes with column inconsistencies

To get design support in any of these design areas, select the appropriate type of design support from the **Designer** pull-down of the Index notebook, and then select the desired action.

Assigning the Index to a Physical Design

Only, when an object is assigned to a physical design, you can fully exploit the design support functions available for the entire physical design. For example, from the physical design, you can request design proposals for several objects at a time.

To assign the index object to a physical design:

1. Open the Index notebook to the Designs page.
2. Click **Search** to search for an appropriate physical design among the list of existing physical designs.

Defining the Index

To specify the parameters by which the index is defined:

1. Open the Index notebook to the Definition page.
2. Specify the listed parameters:
   - ⇑ Owning table

     Click **Open** to open the notebook of the owning table.

     Click **Search** to search for the table on which you want to define the index.
   - ⇑ Password
   - ⇑ Subpages
   - ⇑ Buffer pool
   - ⇑ Defer
   - ⇑ Close
   - ⇑ Organization
     - Unique
     - Clustered
       - Partitioned
         - Number of partitions

Note:

You cannot specify a partitioned index if the owning table is not partitioned.

Specifying Index Columns

To specify the columns on which you want to define the index:

1. Open the Index notebook to the Columns page.
2. From the **Available Columns** list box, select the columns on which you want to define the index.
3. Click **Add** to add the selected columns to the index.

   The selected columns are added to the **Selected Columns** list.
4. Click **Remove** if you want to remove selected columns from the index.

   The selected columns are dropped from the **Selected Columns** list.

Specifying the Sorting Order

To specify the sorting order of index columns:

1. Open the Index notebook to the Columns page.
2. From the columns in the **Selected Columns** list, select a column.

   The type of sorting order of this column is indicated. It can be either of the following:
   ⇑ Ascending
   ⇑ Descending

3. Change the sorting order, if needed.

Specifying the Storage Information

To specify the storage information of the index:

1. Open the Index notebook to the Storage page.
2. Specify the listed storage information.

Viewing Index Partitions

You can only view the index partitions on this page. To create an index partition, switch to the notebook of the owning table and request a proposal for the creation of indexes.

Note:

Only, when the table is partitioned, you can get a proposal for a partitioned index.

To view the existing index partitions:

1. Open the Index notebook to the Partitions page.
2. In the Partition definition list, you see how many index partitions exist, and for which columns they are defined.
3. In the Partitions list, you see the type and storage information of each partition.

Click **Open** to open the Index Partition window, where you can specify the type and the storage information of each partition .

Specifying Type and Storage Information of an Index Partition

To specify the type and the storage information of index partitions:

1. Open the Index notebook to the Partitions page.
2. From the **Partitions** table, select the partition for which you want specify the type and the storage information.
3. Click **Open** .

   The Index Partitions window is displayed.
4. Fill in the listed type and storage information.
5. Close the window.

   You see the specified values in the **Partitions** table. The **Type** column shows whether you selected to manage the data sets for the index yourself (VCAT option), or whether you want DB2 to do this for you (Storage Group option).

Viewing DB2 Actual Values

To view the index statistic values from the DB2 catalog:

⇑ Open the Index notebook to the DB2 Actuals page.

The current values of the index or of the index partitions are displayed.

The index statistical values are extracted from the DB2 catalog through extraction of the values of the owning table.

No data is available for extraction as long as the owning table object has not been implemented.

Designing a Table Space Object

The physical design is carried out with notebooks. The notebook sections indicate the different design areas.

You perform the design by filling in the necessary fields on the notebook pages. In addition, DataAtlas Designer offers design support. For the table space object, all types of design support are available:

⇑ Information on potential design problems
⇑ Design proposals
⇑ Design validations

The design of the table space object consists of the following tasks:

⇑ Specifying general information
⇑ Assigning the table space to a database
⇑ Assigning the table space to a physical design
⇑ Setting options
⇑ Creating table space partitions
⇑ Changing type and storage information of a table space partition
⇑ Specifying storage information
⇑ Assigning tables
⇑ Selecting a buffer pool
⇑ Specifying design information
⇑ Extracting DB2 actual values

Getting Design Support for a Table Space

You can get design support from anywhere in the notebook.

For the table space object, the following design support is available:

⇑ Inform - Information on potential design problems:
- -Identification of table spaces with default values
- -Identification of table spaces with design inconsistencies

⇑ Propose - Design proposals:
- -Calculation of space requirements
- -Calculation of SEGSIZE
- -Setting of options

⇑ Validate - Design validations:
- -Identification of invalid table spaces assigned to DSNDB07
- -Identification of invalid table spaces
- -Identification of incomplete table spaces

To get design support in any of these design areas, select the appropriate type of design support from the **Designer** pull-down of the Table Space notebook, and then select the desired action.

Assigning the Table Space to a Database

To assign the table space to a database object:

1. Open the Table Space notebook to the General page.
2. Click **Search** beside the database entry field to search for an existing database to which you can assign the table space.

Assigning the Table Space to a Physical Design

Only, when an object is assigned to a physical design, you can fully exploit the design support functions available for the entire physical design. For example, from the physical design, you can request design proposals for several objects at a time.

To assign the table space object to a physical design:

1. Open the Table Space notebook to the Designs page.
2. Click **Search** to search for an appropriate physical design among the list of existing physical designs.

Setting Options

To set the options for a table space object:

1. Open the Table Space notebook to the Options page.
2. Specify the following:
    - ⇑ Type
        - -Simple
        - -Segmented
            - -Segment size
        - -Partitioned
            - -Number of partitions
    - ⇑ LOCKSIZE
    - ⇑ Password
    - ⇑ Close option

Creating Table Space Partitions

To create table space partitions:

1. Open the Table Space notebook to the Partitions page.
2. Click **Create** .

   The Table Space Partitions window opens.
3. Specify the running number of the partition that you want to create.

   You can create as many partitions as you specified on the Options page.
4. Specify the necessary type and storage information.
5. Click **OK** .

   You are back on the Partitions page and, in the Partitions table, you see the newly created table space partition with the values you specified.

Note: With a partitioned table space, you must specify a partitioned index for the table that is assigned to the table space.

If you do not specify the information on using block, free block, or compress for a partition, the table space values as specified on the Storage page also hold for the partition.

Specifying Type and Storage of a Table Space Partition

To specify the type and the storage information of a table space partition:

1. Open the Table Space notebook to the Partitions page.
2. From the **Partitions** table, select the partition whose information you want to specify.
3. Click **Open** .

   The Table Space Partitions window opens.
4. Change the necessary type and storage information.
5. Click **OK** .

   You're back on the Partitions page and the values you specified are listed in the **Partitions** table.

Note: If you do not specify the information on using block, free block, or compress for a partition, the table space values as specified on the Storage page also hold for the partition.

Specifying Storage Information

To specify the storage information for the table space object:

1. Open the Table Space notebook to the Storage page.
2. Specify the necessary type and storage information.

Assigning Tables

To assign tables to the table space object, do the following:

1. Open the Table Space notebook to the Tables page.
2. Click **Search** to search for existing tables to assign to the table space.
3. Click **Remove** to remove selected tables from the present table space.

The assigned tables are listed in the list box.

Selecting a Buffer Pool

To select a buffer pool for the table space, do the following:

1. Open the Table Space notebook to the Buffer Pool page.
2. From the list box, select the appropriate buffer pool for the table space.

   The selected buffer pool is listed in the **Buffer pool** entry field.

Specifying Design Information

To specify the design information for a table space object:

1. Open the Table Space notebook to the Design Info page.
2. Specify the following values:
   - Keep in main storage
   - Usage intent

Extracting DB2 Actual Values

Here you extract the most important DB2 statistic values of the table space from the DB2 catalog. Together with the values of the table space, the statistic values of the assigned storage group are retrieved.

To extract the statistic values of the table space object:

1. Open the Table Space notebook to the DB2 Actuals page.
2. Click **Search** to search for the database that contains the DB2 catalog from which you want to extract the statistic values.
3. Click **Extract statistics** to extract the DB2 statistic values of the table space from the DB2 catalog.

   The values are listed in the corresponding fields.

Note: No data is available for extraction as long as the table space has not been implemented.

Designing a Database Object

The physical design is carried out with notebooks. The notebook sections indicate the different design areas.

You perform the design by filling in the necessary fields on the notebook pages. In addition, DataAtlas Designer offers design support. For an existing database object, you can get design support in terms of Information on potential design problems.

The design of a database object consists of the following tasks:

⇑ Specifying general information
⇑ Setting options
⇑ Specifying the design information
⇑ Viewing assigned tables spaces
⇑ Assigning the database to a storage group
⇑ Selecting a buffer pool
⇑ Assigning the database to a physical design

Getting Design Support for a Database

You can get design support from anywhere in the notebook.

For the database object the following design support is available:

⇑ Inform - Information on potential design problems:
-Identification of databases with default values
-Identification of databases with performance problems

No Propose or Validate functions are available for the index object.

To get design support , select **Inform** from the **Designer** pull-down of the Database notebook.

Setting Options

To set the options for a database object:

1. Open the Database notebook to the Options page.
2. Specify the listed options.

Specifying the Design Information

To specify the design information of a database object:

1. Open the Database notebook to the Design Info page.
2. Specify the listed design information .

Viewing Assigned Table Spaces

To view the table space objects that are assigned to the database object:

1. Open the Database notebook to the Table Spaces page.
2. In the **Assigned table spaces** list box, you see the table spaces that are currently assigned to the database.

   Click **Open** to open the notebook of the selected table space.

You cannot assign table spaces to the database on this page. To assign a table space to a database, you need to open the notebook of the table space.

Assigning the Database to a Storage Group

To assign the database to a storage group:

1. Open the Database notebook to the Storage Group page.
2. Click **Search** to search for an existing storage group to which you can assign the database.

Selecting a Buffer Pool

To select a buffer pool for the database object:

1. Open the Database notebook to the Buffer Pool page.
2. From the list, select the appropriate buffer pool for the database.

   The selected buffer pool is listed in the **Buffer pool** entry field.

Assigning the Database to a Physical Design

To assign the database to a physical design:

1. Open the Database notebook to the Designs page.
2. Click **Search** to search for an existing physical design to which you can assign the database.

Designing a Storage Group Object

The physical design is carried out with notebooks. The notebook sections indicate the different design areas.

You perform the design by filling in the necessary fields on the notebook pages. In addition, DataAtlas Designer offers design support. For the storage group object, all types of design support are available:

⇑ Information on potential design problems
⇑ Design proposals
⇑ Design validations

The design of the storage group object consists of the following tasks:

⇑ Specifying general information
⇑ Assigning the storage group to a physical design
⇑ Specifying storage information
⇑ Assigning databases
⇑ Viewing assigned table spaces
⇑ Viewing assigned indexes
⇑ Specifying the usage intent
⇑ Specifying the required space
⇑ Converting the used space value
⇑ Viewing DB2 actual values

Getting Design Support for a Storage Group

You can get design support from anywhere in the notebook.

For the storage group object, the following design support is available:

⇑ Inform - Information on potential design problems:
 -Identification of storage groups with SMS usage

⇑ Propose - Design proposals:
 -Calculation of required space

⇑ Validate - Design validations:
 -Identification of storage groups with invalid definitions
 -Identification of incomplete storage groups

To get design support in any of these design areas, select the appropriate type of design support from the **Designer** pull-down of the Storage Group notebook.

Assigning the Storage Group to a Physical Design

To assign the storage group to a physical design:

1. Open the Storage Group notebook to the Designs page.
2. Click **Search** to search for an existing physical design to which you can assign the storage group.

Specifying Storage Information

To specify the storage information of a storage group object:

1. Open the Storage Group notebook to the Storage page.
2. Enter the volume catalog (VCAT) name in the provided entry field.
3. Specify whether the volume catalog is password protected, and enter the password, if necessary.
4. Specify the kind of device type.
5. Select one of the following types of storage management:
   - ⇑ Storage management by the system (SMS)
   - ⇑ Storage management by the user

If you decide to manage the storage yourself, specify the volumes:

1. Click **Add volume** to add the volume specified in the entry field to the list of available volumes.
2. Click **Delete volume** to delete the volume specified in the entry field from the list of available volumes.
3. Click **Add** to add selected volumes from the list of available volumes to the list of volumes selected for allocation.
4. Click **Remove** to remove selected volumes from the list of volumes selected for allocation.

Assigning Databases

To assign databases to a storage group object:

1. Open the Storage Group notebook to the Databases page.
2. Click **Search** to search for existing databases to assign to the storage group.
3. Click **Remove** to remove selected databases from the present storage group.

The currently assigned databases are listed in the list box.

Viewing Assigned Table Spaces

To view the table space objects assigned to a storage group object:

⇑ Open the Storage Group notebook to the Table Spaces page.

You see a list with the table spaces currently assigned to the storage group.

You cannot assign any table spaces to the storage group on this page. To assign a table space to the storage group, you need to open the notebook of the table space.

Viewing Assigned Indexes

To view the index objects assigned to a storage group object:

1. Open the Storage Group notebook to the Indexes page.

   You see a list with the indexes currently assigned to the storage group.

You cannot assign any indexes to the storage group on this page. To assign an index to the storage group, you need to open the notebook of the index.

Specifying the Usage Intent

To specify the usage intent of the storage group:

1. Open the Storage Group notebook to the Design Info page.
2. Specify one of the following:
   - ⇑ Table spaces (general)
   - ⇑ Indexes (general)
   - ⇑ Table spaces
   - ⇑ Indexes

     Click **Search** to search for the appropriate table space or index.
   - ⇑ None

Specifying the Required Space

To specify the space required by the storage group:

1. Open the Storage Group notebook to the Design Info.
2. Fill in the entry field for required space.
3. Click **Convert** to convert the required space value from kilobytes to the number of cylinders or tracks.

Note:

On request, DataAtlas Designer calculates the required space value and offers a proposal.

For an optimum proposal you need to have assigned the necessary table spaces and created the necessary indexes. Also, the following table data load specifications are needed:

⇑ Initial number of rows
⇑ Confidence factor

Converting the Used Space Value

The amount of space used by a storage group is given in terms of kilobytes.

To see this value converted from kilobytes to the number of cylinders or tracks:

1. Open the Storage Group notebook to the DB2 Actuals page.
2. Click **Convert.**

   The Conversion Utility window opens.
3. Select the appropriate device type.

   You see the used space value converted from kilobytes to the number of cylinders or tracks.

Viewing DB2 Actual Values

To view the storage group statistic values from the DB2 catalog:

⇑ Open the Storage Group notebook to the DB2 Actuals page.

The current values are displayed.

The storage group values are extracted through extraction of the values of one of the table spaces assigned to the storage group.

If the table spaces have not been implemented yet, no extraction data is available.

Designing a View Object

The physical design of the database design objects is carried out with notebooks. The notebook sections indicate the design areas of an object.

The design of the view object consists of the following tasks:

⇑ Specifying general information
⇑ Defining a view

Defining a View

To define a view object:

1. Open the View notebook to the Definition page.
2. Specify the following item:
    - ⇑ SQL select statement (Subselect)
    - ⇑ Check option
    - ⇑ Comment
    - ⇑ Label

3. Optionally, you can list the participating Columns of the View .

Creating a DB2/390 Alias or Synonym Object

To create a DB2/390 alias or synonym object, first follow the usual steps for creating objects .

An empty notebook for the newly created alias or synonym object opens.   Its sections require the following tasks:

⇑ Specifying general information
⇑ Defining an alias or synonym

Defining an Alias or Synonym

To define an alias or synonym object:

1. Open the Alias or Synonym notebook to the Definition page.
2. Specify one of the following:
   - ⇑ Synonym
   - ⇑ Alias
     - Comment
     - Label

3. Your choice of either synonym or alias determines your further specification:

   Synonym (or Alias) for
   ⇑ Table
   ⇑ View

   Click **Search** to get a list of table or view objects that can be entered into the corresponding field. If you select an object from the list, it will appear in the field.

   Click **Clear** to clear the corresponding entry field so that you can search for another object.

Dictionary Tasks

The main tasks you can perform with the Dictionary are:

- Creating objects
- Populating Data Definitions
- Updating objects
- Generating Definitions
- Querying the TeamConnection Database
- Deleting Objects

Creating Objects

There are two basic reasons for creating objects with DataAtlas:

⇑ You need a data definition that can't be acquired by populating one that already exists.

⇑ An object requires the existence of another object, so you have to create the required object first.

The first object you create is a workfolder. It's a special object whose menu bar contains the **Create object** action -- the action you use to create anything else. You also need a workfolder to update any other object, because an object must be represented in a workfolder before it can be changed.

Related Topics

⇑ Steps for Creating a Workfolder
⇑ Required Objects

Steps for Creating a Workfolder

To create a workfolder:

1. In the **Main Folder** window, double-click the **Workfolders** icon.

   The **Workfolders** folder opens.

2. Click **Folder** from the menu bar; then click **Create workfolder** .

   The **Workfolder** notebook opens.

3. Give the workfolder a name in the **Name** field, and make sure the fields on the **TeamConnection** page are filled in; then click **OK** .

   An icon representing your new workfolder appears in the **Workfolders** folder, and your workfolder opens, ready for use.

If you want to practice these steps, try Lesson 1.1 of the tutorial. Then try the other steps in Lesson 1 so that you can see how a workfolder is used to create other objects.

Required Objects

Before you can populate a DB2/390 table or a DB2 UDB table, you must create a relational system object. For DB2/390, this is the name of a system; for DB2 UDB, this is the name of a server. To create one, you select **Workfolder** and then **Create object** from your workfolder's menu bar, select "DB2 System" in the Create Object window, and specify a system and creator in the DB2 System notebook.

DB2 UDB tables also require a database object. To create one, again select **Create object** from your workfolder, but this time select "DB2 UDB Database" from the **Create Object** window. Then specify a system name and database name in the DB2 UDB Database notebook.

Populating Data Definitions

Populating is the task of importing data definitions into the TeamConnection database. You can populate DB2/390 table definitions, DB2 UDB table definitions, IMS DBDs, IMS PSBs, COBOL COPY files, and PL/I include files.

DataAtlas populates DB2 tables from wherever they reside, via the relational catalog. It also populates the definitions of other objects the tables need - storage groups, table spaces, indexes, views, synonyms, or aliases.

DataAtlas populates IMS DBDs and PSBs after they have been downloaded from MVS to a workstation file. COBOL COPY files and PL/I include files are also populated from workstation files.

When you populate a data definition, you can choose to reconcile its local data components the same time. If you do, DataAtlas refers to a mapping table and takes one of the following actions for each local data component that is mapped to a shareable data component in the table:

⇑ If the shareable data component is stored in the TeamConnection database, DataAtlas assigns to the local data component the attributes of the shareable data component.

⇑ If the shareable data component is not stored in the TeamConnection database, DataAtlas creates a shareable data component there and gives it the attributes of the local data component.

Related Topics

⇑ Steps for Populating
⇑ Creating a Mapping Table from a Prototype
⇑ Populate Reports
⇑ Specifying a Relational and Physical Design
⇑ Creating Shareable Data Components without Reconciling
⇑ When Populating Affects DBD Specifications
⇑ Limitations When Populating PL/I and COBOL Files

Steps for Populating

To populate data definitions of DB2 tables, IMS DBDs and PSBs, COBOL COPY files, and PL/I include files:

1. In the **Main Folder** window, double-click the **Populate from** icon.
2. Select a folder from the **Populate from** folder.
3. Bring up the pop-up menu for the selected folder.
4. Select a populate option.

   The appropriate Search window opens.
5. Specify in the Search window the data definitions you want to populate and click **Continue** .

   The Populate notebook opens.
6. Fill in the fields of the Populate notebook.

   Decide if you want DataAtlas to reconcile the data definitions based on a mapping table . If you do, check the **Reconcile** box and specify the mapping table in the **Use mapping table** field.
7. Click **Populate** .

   The populate operation begins. When it finishes, DataAtlas displays a populate report.

If you want to practice these steps with sample data, try Lesson 2 of the tutorial.

Creating a Mapping Table from a Prototype

If you use a mapping table that contains no entries for some of the local data components, DataAtlas creates a prototype mapping table for them. The created table is called a prototype because the source names and the target names are the same; you'll probably have to edit the prototype to get the target names you want.

To create a prototype mapping table that represents all the local data components you want to populate and reconcile, you have to be sure of two things:

⇑ The populate operation that creates the prototype mapping table must be a **trial run** ; that is, nothing must be actually populated.

⇑ The mapping table used in the trial run must contain none of the local data components you eventually want to populate; otherwise, the prototype mapping table will have no entries for these components. For this purpose, you can use the sample mapping table that's shipped with DataAtlas.

Here are the steps to follow:

1. In the **Populate** notebook, check the **Reconcile** box.
2. In the **Use mapping table** field, enter the path and file name of the sample mapping table:

```
&lt.DataAtlasInstallPath>\lang\en_us\samples\datatlas.map
```

   Where `&lt.DataAtlasInstallPath>` is the root directory where DataAtlas resides is installed.

3. Click **Trial Run** .

   Result: No objects are stored in the TeamConnection database. The populate operation produces a report and a prototype mapping table named *dddnnnnn* `.MAP` .

   ddd designates the kind of data definition you processed: RDB (relational database), DBD, PSB, COB, or PLI.

   nnnnn is a generated hexadecimal number.

4. Close the report window and cancel out of the **Populate** notebook.
5. Edit the prototype mapping table to reflect the target names you want for your shareable data components. (You'll find the prototype mapping table in the directory specified on the **General** page of your **Profile** notebook.)

Populate Reports

DataAtlas produces a report at the end of every populate operation. The report is stored on the drive and directory you specify on the General page of the **Profile** notebook.

The file names of populate reports depend on the types of objects you populate, as shown in this table:

Object Type	File Name
DB2 table	RDB *nnnnn* .RPT
DBD	DBD *nnnnn* .RPT
PSB	PSB *nnnnn* .RPT
COBOL COPY files	COB *nnnnn* .RPT
PL/I include files	PLI *nnnnn* .RPT

The ***nnnnn*** is a generated hexadecimal number. Because file names are unique, they will accumulate over time. You'll want to delete old files periodically.

Specifying a Relational and Physical Design

When you populate DB2 tables, you can specify that the populated objects - the tables and their associated objects - be associated with a named relational design, with a named physical design, or with both. This enables you to later identify the relational design to DataAtlas Modeler and work with the objects collected within that relational design. Similarly, you can identify the physical design to DataAtlas Designer and work with the objects collected within it.

Creating Shareable Data Components without Reconciling

When you populate a DB2 table, you can ask DataAtlas to treat every column as if it were specified in a mapping table, even though you're not using a mapping table. To do so, you check the **Use or create data elements for all columns** check box. If you use a mapping table **and** this check box, you're assured of creating shareable data elements for columns that aren't in the mapping table.

When you populate a COBOL or PL/I data structures, even if you do not request reconcile processing, DataAtlas treats group objects in the file as if you did request it. That is, DataAtlas creates shareable data structures in the TeamConnection database and gives them the attributes of the group objects.

When Populating Affects DBD Specifications

When you populate a DBD, the specifications in it are stored without changes in all but two cases:

⇑ If the device is specified as 2314, DataAtlas changes it to 2319, a similar device type.

⇑ If the access method is specified as ISAM, DataAtlas changes it to VSAM.

Limitations When Populating PL/I and COBOL Files

COBOL REDEFINES attributes aren't carried forward when you populate source files. This limitation is also true of a number of PL/I attributes; these are identified in Appendix E of *DataAtlas Dictionary and Designer User's Guide* . In addition, PL/I variables that are implicitly or contextually defined aren't supported by the Populate function.

Even though you can't populate these attributes, you can specify values for them in the Included Source Definition notebook. When you generate COPY and include files, they will contain the values you specified.

Updating Objects

To update an object in the TeamConnection database it must be represented in your workfolder. If you created the object you want to update, its icon is already in your workfolder. But if another user created it, or if it was populated into the TeamConnection database, you have to search for it to get its icon into your workfolder.

By double-clicking on an icon in your workfolder, you open the settings notebook for the object. If you modify any setting in the notebook and select the **OK** or **Save** button, you update the object.

Related Topic

⇑ Steps for Searching

Steps for Searching

To search for an object in the TeamConnection database:

1. In the **Main Folder** window, double-click the **Workfolders** icon.

2. In the **Workfolders** folder, double-click a workfolder icon to open the workfolder.

3. Click **Workfolder** on the menu bar of the workfolder; then click **Search TeamConnection database** .

   The TeamConnection Database - Search notebook opens.

4. Select the object types you want to search for. Decide whether to qualify the search by specifying an object name, variation, or revision. If you specify no qualifications, the search will find all the object instances of the types you selected, within the family, release, and work area assigned to the workfolder.

5. Click **Search** .

   When the search finishes, the TeamConnection Database - Search Results window opens. In it are the names of all the objects that matched your search criteria.

6. Select the objects you want represented in the workfolder; then click **OK** . An icon for each selected object appears in the workfolder.

If you want to practice these steps, try Lesson 3.1 of the tutorial. Then try the other steps in Lesson 3 so that you can practice the updating task on two sample objects.

Related Topic

⇑ DataAtlas Naming Scheme

DataAtlas Naming Scheme

Each DataAtlas object has a multipart name of the form:

⇑ &lt. prefix > access-name &lt. variation : revision >

The names of objects can have additional qualifiers to help you make them unique. The qualifiers precede the access name and indicate a hierarchy in which the object exists. You can see a hierarchy of qualifiers in the second example below. PAYROLL_TABLE, a table object, has a **system** qualifier (m78serv3), a **creator** qualifier (amy), and **database** qualifier (servsamp), in that order. Notice that the qualifiers are separated from each other by a colon. See Appendix B of *DataAtlas Dictionary User's Guide* to learn which qualifiers you can use with which objects.

The only part of a name that is required is the access name, which you choose and maintain. The full length of the name, qualifiers and special characters included, can be up to 255 characters. No limit exists on any part of a name, but when you choose an access name, variation, and revision, take into account the length of the other qualifiers.

Examples

⇑ **IMS DBD:**
⇑ &lt.>BE3ORDER&lt.V1:R1>
⇑
⇑ **IMS PSB:**
⇑ &lt.>PE3ORDER&lt.:>
⇑
⇑ **IMS PCB:**
⇑ &lt.>PE3ORDER::PCB1&lt.:>
⇑
⇑ **DB2 UDB Table:**
⇑ &lt.>m78serv3:amy:servsamp::PAYROLL_TABLE&lt.:>
⇑
⇑ **Included Source Definition:**
⇑ &lt.>ABC.CPY&lt.:>
⇑
⇑ **Shareable Data Structure:**
⇑ &lt.>EMPLOYEE_REC&lt.V1:R1>
⇑
⇑ **Shareable Data Element:**
⇑ &lt.>ZIPCODE_9&lt.V1:R0>
⇑
⇑ **Data Element Alias:**
⇑ &lt.ZIPCODE_9>zipcode&lt.:>

Generating Data Definitions

Generating is the task of exporting data definitions from the TeamConnection database to a workstation file.   You perform this task after you have:

⇑ Created an object containing a new data definition
⇑ Updated an object containing an existing data definition

When you generate a relational database definition, the file contains DDL that you can submit to the appropriate DB2 or Oracle subsystem. An alternative to generating DDL for many individual definitions is to generate a Included source definition. In this case, the DDL for all the definitions in the physical design is generated to the file.

When you generate an IMS DBD or PSB, the file contains IMS macro statements that you can upload to the appropriate MVS system.

When you generate an included source definition, the file contains source code that you can include in a COBOL or PL/I program and compile with the appropriate compiler.

You can generate data definitions in two ways:

⇑ From the DataAtlas user interface
⇑ By using TeamConnection's build function and The DATATLAS.EXE build script

Related Topic

⇑ IMS, COBOL, and PL/I Validation Reports

Generating from the DataAtlas User Interface

To generate a data definition from the DataAtlas user interface, follow this procedure:

1. Open a workfolder that contains an icon for the object whose definition you want to generate.

2. Click the icon with the right mouse button.

   A menu opens.

3. Click **Generate to file** .

   The **Generate to File** notebook opens to a page for DB2 definitions , Oracle definitions , physical designs , IMS definitions , or included source definitions .

4. Decide whether to use the default file name or specify your own, and select from the other notebook options.

5. Click **Generate** .

   The data definition is generated to the specified file.

If you want to practice these steps with a sample data definition, try Lesson 4.1 of the tutorial.

The DATATLAS.EXE Build Script

An executable file named DATATLAS.EXE is delivered with DataAtlas. This executable can be used by TeamConnection as a binary build script to generate data definitions. It can also be invoked from within another build script that you have written to better meet your environmental or user-specific needs.

You can use DATATLAS.EXE to generate the following output: DB2 UDB DDL, DB2/390 DDL, Oracle DDL, IMS DBD and PSB source, COBOL COPY files, and PL/I include files.

The DATATLAS.EXE build script is in the subdirectory `&lt.DataAtlasInstallPath>\client` in OS/2 and in `&lt.DataAtlasInstallPath>\bin` in Windows NT.

Related Topic

⇑ Steps for Doing a Build

Steps for Doing a Build

After your TeamConnection administrator has set up a TeamConnection build server, follow these steps:

1. Create a builder for the DATATLAS.EXE build script.

2. Create an empty file part to receive the output from the build, and associate the builder with it. This output file part is empty only until you invoke the first build. Subsequent builds will overlay its contents.

3. Connect the parts you want to build to the output file part, creating a build tree with the output file part at the top of the tree.

4. Build the output file part at the top of the build tree.

For more information about the TeamConnection build process, see the *TeamConnection User's Guide*, SC34-4499. For information about the input parameters to use when creating a builder, the return codes from the DATATLAS build script, and the format of the input file it passes to DataAtlas, refer to Appendix C of *DataAtlas Dictionary User's Guide*, SC26-9039.

IMS, COBOL, and PL/I Validation Reports

DataAtlas produces a validation report whenever it detects an error in the generation of a DBD, PSB, COBOL COPY file, or PL/I include file. The reports are named VAL *nnnnn* .RPT; the *nnnnn* is a generated hexadecimal number.

The reports are stored on the path where your populate reports are stored. Because the file names are unique, they will accumulate over time. You'll want to delete old files periodically.

Querying the TeamConnection Database

To query the TeamConnection database, you can write your own SQL query or use an existing query.

Many queries are supplied with DataAtlas. You can practice with them by querying sample data, and you can modify them to suit your needs. You'll find them:

⇑ In your SQL Query folder

⇑ On the path `&lt.DataAtlasInstallPath>\lang\en_us\samples\query` , where `&lt.DataAtlasInstallPath>` is the root directory where DataAtlas is installed

Chapter 8 of *DataAtlas Dictionary User's Guide* gives their file names, classifies them by type, and identifies the objects that are their targets. This chapter also tells you about:

⇑ The SQL notational simplifications that the data model for the TeamConnection database offers for query writing

⇑ TeamConnection's named views and attributes for the object types that can be stored in its database

Both subjects are useful when you want to write your own queries.

Related Topic

⇑ Steps for Querying

Steps for Querying

To query the TeamConnection database:

1. In the **Main Folder** window, double-click the **SQL Query Folder** icon.

   The SQL Query Folder opens.

2. Click **Query** on the menu bar; then click **Create SQL Query** .

   An icon for a new query appears in the folder, and a properties notebook for the query opens to the General page .

3. If you want to write your own query, name it and go to the Query page . If you want to use an existing query that has a supplied-query format , click **Search** , and use the **Query File - Search** window to find the file that contains it. After you have selected the file, tab to the **Query** page.

4. Type in your query or modify the existing query to suit your needs. Then click **Run** .

   The query runs. When it finishes, an SQL Report notebook opens to the **Report** page, where the results of the query are displayed. On the General page of the notebook, you can name the report and store it in a workstation file.

If you want to practice these steps with sample data, try Lesson 5.1 of the tutorial.

Deleting Objects

In DataAtlas there are two kinds of deletions: *iconic* delete and *original* delete.

An iconic delete doesn't affect objects in the TeamConnection database; only the icon representing an object is deleted from the folder containing it. For example, you can do an iconic delete of objects in a workfolder. Select one or more icons there; then click **Selected** on the menu bar. One of the actions you'll see on the menu is **Delete** . This is iconic delete. Select it and the icons disappears from the workfolder.

On the same menu is **Original** , meaning original delete. If you click it, a procedure begins for deleting the objects from the workfolder **and** from the TeamConnection database.

Original delete is an action to consider carefully. The object you choose to delete might be related to many other objects -- as a container for them, for example. The original-delete procedure lets you see what the related objects are and then either selectively delete them or cancel the entire procedure.

Before you delete an object, consider running a supplied query to see what impact the deletion would have.

Related Topics

⇑ Steps for Original Delete
⇑ Supplied Queries to Use before Deleting

Steps for Original Delete

To delete objects in the TeamConnection database:

1. Open a workfolder that contains icons for the objects you want to delete; select the icons.

2. Click **Selected** on the menu bar.

3. Click **Original** on the menu; then click **Delete** .

   The Delete notebook opens to the Delete page . The objects you selected for deletion appear in the **Objects to be deleted** list. The objects related to the top entry appear in the **Related objects to be deleted** list.

4. Deselect the related objects you do not want to delete.

5. If there is another entry in the **Objects to be deleted** list, select it. When its related objects appear, again deselect the ones you do not want to delete. Repeat this process until you have made a deletion decision about all the objects related to each of the entries in the **Objects to be deleted** list.

6. If you want to see a summary your deletion choices, check the **Confirm on delete** box.

7. Click **Delete** .

   If you didn't ask for a confirmation, the objects that remained selected are deleted. If you did (as is assumed here), the Delete Confirmation window opens.

8. Review your deletion choices in the **Delete Confirmation** window. If they are still what you want, click **Delete** ; otherwise, click **Cancel** .

   All the selected objects are deleted when you click **Delete** . Clicking **Cancel** returns you to the **Delete** notebook.

Supplied Queries to Use before Deleting

The supplied query you choose to show the impact of a deletion depends on the object type you're thinking of deleting. The following table shows which object types correspond to which supplied queries. The supplied queries are identified by both their file names and query names. You'll find them on the path On the path `&lt.DataAtlasInstallPath>\lang\en_us\samples\query` , where `&lt.DataAtlasInstallPath>` is the root directory where DataAtlas is installed.

Object Type	File Name	Query Name
Data element alias	EWSQDEAU.QRY	DA_DataElementAliasUsage
Data structure alias	EWSQDSAU.QRY	DA_DataStructureAliasUsage
Creator/schema	EWSQROC.QRY	DA_RelationalCreatorContents
DB2 system	EWSQRSC.QRY	DA_RelationalSystemContents
DB2/390 alias/synonym	EWSQRA.QRY	DA_RelationalAlias
DB2/390 buffer pool	EWSQDMBP.QRY	DA_DB2390BufferpoolUsage
DB2/390 database	EWSQDMDC.QRY	DA_DB2390DatabaseContents
DB2/390 ICF catalog	EWSQMICU.QRY	DA_MVSICFCatalogUsage
DB2/390 index	EWSQDMIG.QRY	DA_DB2390IndexGlossary
DB2/390 index space	None	
DB2/390 physical design	EWSQDMPD.QRY	DA_DB2390PhysicalDesignContents
DB2/390 storage group	EWSQDMSG.QRY	DA_DB2390StoragegroupUsage
DB2/390 table	EWSQDMFK.QRY	DA_DB2390ForeignKey
DB2/390 table space	EWSQDMTS.QRY	DA_DB2390TablespaceGlossary
DB2/390 view	EWSQDMVG.QRY	DA_DB2390ViewGlossary
DB2/390 volume	EWSQMVU.QRY	DA_MVSVolumeUsage
DB2 UDB database	EWSQD2DC.QRY	DA_DB2csDatabaseContents
DB2 UDB index	EWSQD2IG.QRY	DA_DB2csIndexGlossary
DB2 UDB physical design	EWSQD2PD.QRY	DA_DB2csPhysicalDesignContents
DB2 UDB table	EWSQD2FK.QRY	DA_DB2csTableForeignKeys
DB2 UDB table space	EWSQD2TC.QRY	DA_DB2csTablespaceContents
DB2 UDB view	EWSQD2VG.QRY	DA_DB2csViewGlossary
Included source definition	EWSQISG.QRY	DA_IncludedSourceGlossary
IMS DBD	EWSQDU.QRY	DA_DBDUsage
IMS PCB	EWSQPU.QRY	DA_PCBUsage
IMS PSB	None	
Key definition	None	
Oracle index	EWSQDOIG.QRY	DA_OracleIndexGlossary
Oracle table	EWSQDOFK.QRY	DA_OracleForeignKeys
Oracle table space	EWSQDOTC.QRY	DA_OracleTablespaceContents
Oracle view	EWSQDOVG.QRY	DA_OracleViewGlossary
Relational design	EWSQRDC.QRY	DA_RelationalDesignContents
Shareable data element	EWSQDEU.QRY	DA_DataElementUsage
Shareable data structure	EWSQDSU.QRY	DA_DataStructureUsage
Shareable table definition	EWSQTDU.QRY	DA_TableDefinitionUsage
View definition	None	

Windows and Notebooks

This section contains explanations of the DataAtlas windows and notebooks. The explanations are organized alphabetically by window and notebook name.

Note: The headings for frequently occurring menu-bar items, fields, and push buttons do not always appear in these contents under the headings for the windows and notebooks that contain them. However, for every menu-bar item, field, and push button named in a help window, there is a hypertext link from it to an explanatory window.

Action Notebook

Each design action contains several rules.

Use this notebook to tailor an action's set of rules and set the parameters of a rule.

Pages

- Overview of rules
- Details of a rule

Tasks

- Selecting rules
- Setting parameters of rules

Push Buttons

- Get report
- OK
- Cancel
- Help

Overview of Rules

The **Rules** page of the action's properties notebook shows the selected all of the rules that are available for the action.

By default, all the rules are selected.

Deselect the rules you do not want to include in the action. The selected rules are performed when you request a report for the corresponding action.

You find detailed descriptions of the rules on the following notebook pages.

Details of a Rule

The rules listed on the Rules page of the action's properties notebook, are identified by number. These numbers correspond to the tabs on the following notebook pages.

On a numbered notebook page, you find a detailed description of the corresponding rule. A rules is applied when you request the corresponding action by requesting a report.

In some cases, a rule contains parameters. Change the parameters where necessary.

Actions Window

In this window, you see the actions that are available for the type of design support you have selected for a specific type of objects. For example, you see all of the actions available for getting design proposals for one or more tables. Or, you see all of the actions available for getting information on potential design problems for the design of indexes.

Tasks

- Tailoring design support actions
- Performing design support actions

Push Buttons

- Rules
- Get report
- Cancel
- Help

Menu-Bar Choices

With Information and Validation actions:

- Edit
- Window
- Help

With Proposal actions:

- Window
- Help

Edit

This choice displays two options:

Select all	Highlights (selects) the listed items.
Deselect all	Removes highlighting from (deselects) the listed items.

Rules

Click **Rules** to open the notebook for the selected action and see the rules associated with the it. You can see the rules for only one action at a time.

Get Report

Click **Get report** to get the results of performing the selected actions. You can select several actions if you selected either the **Inform** or the **Validate** design support. However, you can select only one action if you selected the **Propose** design support.

Add Objects Window

This window enables you to add objects to a workfolder after the Designer has created new objects, like DB2 indexes. It lists workfolders that are in the same version as the created objects. You select the workfolder you want to add them to.

Lists

List of workfolders
Objects

Push Buttons

OK
Cancel
Help

List of Workfolders

The listed workfolders are in the same version as the new objects. Select one and click **OK** to add the objects to the workfolder.

Objects

This is a read-only list of the objects the Designer created.

OK

Click this button to add the objects to the workfolder you selected from the list. If the workfolder isn't open, it's automatically updated in the TeamConnection database.

Cancel

Click this button if you decided not to add the objects to a workfolder.

COBOL - Populate Notebook

This notebook enables you to:

⇑ Import information about one or more COBOL COPY files into the TeamConnection database

⇑ Perform a trial run to preview the results of this process

The report displayed when a COBOL COPY file has been populated is stored in a file named COB *nnnnn* .RPT. The file is stored in the directory specified on the General page of the Profile notebook.

Pages

- COBOL
- Compiler
- TeamConnection

Push Buttons

- Populate
- Trial run
- Cancel
- Help

COBOL Page

This page enables you to:

⇑ Qualify the names of objects to be populated

⇑ Indicate whether the objects will be reconciled with shareable data components during the populate process

Lists

Files selected
Name

Fields

Variation
Revision
Use mapping table

Check Box

Reconcile

Push Button

Search

Reconcile

Check this box if you want DataAtlas to define populated data structures and data items in terms of shareable data components identified in a mapping table. DataAtlas will create shareable data components if they don't already exist in the TeamConnection database.

Use mapping Table

This field identifies a file that contains the mapping table against which a reconcile operation is performed. Specify the directory only if the file is not in the current directory.

Required?: Only if the **Reconcile** box is checked. Otherwise, this field is disabled.

Compiler Page

This page enables you to specify compiler options by entering them directly or by referring to a command file that contains them.   It also lets you specify a prefix and a suffix to be stripped from a COBOL data name.

If you want to give **SPECIAL NAMES** clauses to the compiler, you can specify them on the **COBOL** page of the **Profile** notebook.

Fields

- Compile options
- Command file
- Prefix
- Suffix

Check Box

- Use command file not compiler options

Push Button

- Search

Compile Options

This field contains the compiler options you want to use in the compilation phase of populating a COBOL COPY file.   Alternatively, you can put compiler options in a command file.

Required?: No.

Default: Initialized from the profile.

Restriction: If you specify additional compiler options, they must not conflict with the default set. For example, you cannot specify NOADATA as a compiler option.

Command File

This field identifies a file that invokes the compiler and passes it compiler options when a COBOL COPY file is populated. Alternatively, you can specify compiler options in the **Compile options** field.

The **Use command file not compiler options** box enables the **Command file** field.

Required?: No.

Default: Initialized from the profile.

Note: The path `&lt.DataAtlasInstallPath>lang\en_us\samples\cobol` contains a sample command file named `ewslpgm.cmd` . `&lt.DataAtlasInstallPath>` is the root directory where DataAtlas is installed.

Prefix

This field can contain a string to strip from the beginning of a COBOL data name. If you enter lowercase alphabetical characters, they will be converted to uppercase before they are used.

Required?: No.

Default: Initialized from the profile.

Suffix

This field can contain a string to strip from the end of a COBOL data name. If you enter lowercase alphabetical characters, they will be converted to uppercase before they are used.

Required?: No.

Default: Initialized from the profile.

Use Command File Not Compiler Options

Check this box if you want DataAtlas to ignore the **Compile options** field and use instead the options in a command file.

When you check this box, the **Command file** field and the **Search** button are enabled.

Default: Initialized from the profile.

Search

Click this button to search for and select the command file you want to use.

Conflicting Objects Window

This window is displayed if:

⇑ You select more than one type of DB2/390 object in a workfolder and

⇑ You select **Inform**, **Validate**, or **Propose** from the Selected menu

This window shows you a list of the object types you selected. Choose one of the types and click the push button for the Designer function you originally requested.

List

Object types

Push Buttons

Inform-Validate-Propose
Cancel
Help

Object Types

The object types you selected in the workfolder are listed here. Select one type and click the push button (**Inform** , **Validate** , or **Propose**) for the Designer function you originally requested.

Inform-Validate-Propose

A push button for the Designer function you originally requested (**Inform** , **Validate** , or **Propose**) is displayed. Click it to process the object type you selected.

Connect to DB2 Window

This window enables you to connect to a relational database system when you're populating or generating relational database objects on a Windows NT workstation.

Fields

- System name
- Logon user ID
- Logon password

Push Buttons

- OK
- Cancel
- Help

System Name

This read-only field contains the name of the database system to which you're connecting. The field is filled in from your selection in the **DB2 Subsystem - Search** window when you're populating and from the **Generate to File** notebook when you're generating.

Logon User ID

Enter a user ID defined on the database system to which you're connecting.

Required?: Only if the database system requires one.

Logon Password

Enter the password associated with the user ID you entered.

Required?: Only if the database system requires one.

OK

Click this button and DataAtlas tries to connect to the database system.

Cancel

Click this button if you decided not to connect to the database system.

Conversion Utility Window

Use this window to view the conversion of values from kilobytes to the number of cylinders or tracks.

This applies to the primary and secondary values of the table space object and to the required and used space values of the storage group object.

Fields

⇑ Quantity (KB):

The value you entered.

⇑ Device:

Depending on the device you choose, DataAtlas Designer calculates the corresponding number of cylinders and tracks.

⇑ Tracks:

The resulting number of tracks.

⇑ Cylinders:

The resulting number of cylinders.

Push Buttons

Click **Cancel** to leave this window.

Click **Help** to see an explanation of this window.

Copy Objects Window

Select the workfolder into which you want the objects copied. When you click **OK**, the target workfolder receives the objects.

If the target workfolder is open, the icons for the objects appear in it. When you close the workfolder, the changes you made are stored. If the target workfolder is closed, DataAtlas copies the objects into it and stores the changes.

Restriction: The target workfolder must belong to the same family, release, and work area as the source workfolder.

List

Objects

Push Buttons

OK
Cancel
Help

Objects

This read-only list identifies the objects that will be copied.

Create Object Window

This window enables you to create DataAtlas objects within the TeamConnection version - the family, release, and work area - you're using. The version is identified on the TeamConnection page of your workfolder's properties notebook.

To see the object type you want to create, you may have to expand one or more of the object groups. (The large Relational Objects group contains subgroups that expand.)

When you see the object type you want, double-click it or click once and click **Create** . The object's properties notebook opens, allowing you to define the new object.

The **Create Object** window remains open until you explicitly close it by clicking **Close** .

Push Buttons

- Create
- Close
- Help

Create

Click this button to open a properties notebook for the object type you selected from the window.

Close

Click this button to close the window.

DataAtlas Main Folder Window

This window is where you begin your work with DataAtlas.   Any task you want to perform originates here.

You can review the tasks DataAtlas supports by selecting from these topics:

- Populating data definitions
- Updating objects
- Generating data definitions
- Querying the TeamConnection database

Menu-Bar Choices

- Folder
- Selected
- Edit
- View
- Window
- Help

Icons

- Populate from...
- Workfolders
- Profile Folder
- Report Folder
- SQL Query Folder

Folder

This choice lets you decide what the folder will look like when it's opened. Its menu has one action, **Open as** , which expands into these options:

Icon view. Shows objects as icons.

Tree view. Shows hierarchic relationships among objects.

Flowed icon view. Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Selected

This choice displays a pull-down menu of the options available for objects that are currently selected.   These are the same options you see if you click on the object with mouse button 2.

Edit

This choice offers two actions:

Select all. Highlights (selects) all the icons in the folder.

Deselect all. Removes highlighting from (deselects) all the icons in the folder.

View

This choice controls the appearance of the icons in the folder. The actions on its menu are:

Icon view. Shows objects as icons.

Tree view. Shows hierarchic relationships among icons.

Flowed icon view. Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Sort. Offers the choice **Name,** which reorders the icons alphabetically by name.

Arrange. Arranges the icons to fit the size of a resized window.

Window

This choice controls the arrangement and availability of DataAtlas windows. It lets you access a window easily when it has been overlaid. The actions on its menu are:

Cascade. Creates an overlapping arrangement of windows. The window from which you selected this action is placed on the bottom.

Minimize all. Minimizes all the windows except the window from which you selected this action.

Maximize all. Maximizes all the open windows. The active window is placed on top of the others.

Restore all. Restores all minimized windows to their previous size.

(Numbered list). A chronological list of windows that are open or minimized. The active window is checked. Clicking another window in the list makes it the active window.

Populate from... Icon

The task of populating data definitions begins with this icon. You can proceed in three different ways:

Double-click it. This opens the **Populate from...** folder, which offers the same options as clicking on the + sign to the left of the icon (see below).

Click it with the right mouse button. This opens a pop-up menu with the options **Open as** and **Help** . **Open as** lets you choose how you want to see the **Populate from...** folder, as an icon view or a tree view. **Help** shows you the same pull-down menu that is under the **Help** menu-bar choice.

Click the + sign. These expansion icons appear:

IMS Source
COBOL Source
PL/I Source
DB2 Source

IMS Source Icon

Double-click this icon to select IMS DBDs to populate.

Click it with the right mouse button to get a pop-up menu with these options:

Help. This displays the same pull-down menu that is under the Help menu-bar choice.

Populate IMS DBD. This lets you select IMS DBDs to populate.

Populate IMS PSB. This lets you select IMS PSBs to populate.

COBOL Source Icon

Double-click this icon to select COBOL COPY files to populate.

Click it with the right mouse button to get a pop-up menu with these options:

Help. This displays the same pull-down menu that is under the Help menu-bar choice.

Populate COBOL. This lets you select COBOL COPY files to populate.

PL/I Source Icon

Double-click this icon to select PL/I include files to populate.

Click it with the right mouse button to get a pop-up menu with these options:

Help. This displays the same pull-down menu that is under the Help menu-bar choice.

Populate PL/I. This lets you select PL/I include files to populate.

DB2 Source Icon

Double-click this icon to select DB2 tables to populate.

Click it with the right mouse button to get a pop-up menu with these options:

Help. This displays the same pull-down menu that is under the Help menu-bar choice.

Populate RDB. This lets you select DB2 tables to populate.

The second option is, in effect, an alternative path to the function offered by double-clicking the icon.

Workfolders Icon

The task of creating a workfolder begins with this icon. You can use it in three ways:

Double-click it. This opens the **Workfolders** folder, which displays the workfolders you have created.

Click it with the right mouse button. This opens a pop-up menu with the options **Open as** and **Help** . **Open as** lets you choose how you want to see the **Workfolders** folder, as an icon view or a flowed icon view. **Help** shows you the same pull-down menu that is under the **Help** menu-bar choice.

Click the + sign. The + sign appears when you have created at least one workfolder. When you click the + sign, icons for your workfolders are displayed in a flowed icon view below the **Workfolders** icon. You can open any of the workfolders by selecting its icon.

Related Topic

Workfolders Folder

Profile Folder Icon

The task of setting up your profile begins with this icon.   You can use it in two ways:

Double-click it. This opens the **Profile** folder, which shows the profiles you have.

Click it with the right mouse button. This opens a pop-up menu with the options **Open as** and **Help**.   **Open as** lets you choose how you want to see the **Profile** folder, as an icon view or a flowed icon view.   **Help** shows you the same pull-down menu that is under the **Help** menu-bar choice.

Related Topic

Profile Folder

Report Folder Icon

You can use this icon in two ways:

Double-click it. This opens the Report folder, which contains all the saved reports that were created during SQL query operations.

Click it with the right mouse button. This opens a pop-up menu with the options **Open as** and **Help**. **Open as** lets you choose how you want to see the **Report** folder, as an icon view or a flowed icon view. **Help** shows you the same pull-down menu that is under the **Help** menu-bar choice.

Related Topic

Report Folder

SQL Query Folder Icon

The task of querying the TeamConnection database begins with this icon. You can use it in two ways:

Double-click it. This opens the **SQL Query** folder, from which you can create and run SQL queries.

Click it with the right mouse button. This opens a pop-up menu with the options **Open as** and **Help**. **Open as** lets you choose how you want to see the **SQL Query** folder, as an icon view or a flowed icon view. **Help** shows you the same pull-down menu that is under the **Help** menu-bar choice.

Related Topic

SQL Query Folder

Database Alias Search Window

This window enables you to select a database to use in generating altered DDL.

List

- Database alias (Database)

Push Buttons

- Continue
- Cancel
- Help

Database Alias (Database)

This list identifies DB2 databases that have been cataloged. Select one to be used in the generate processing; then select the **Continue** push button.

Continue

Click this button to continue generate processing, using the database you selected from the list.

DB2 Subsystem - Populate Notebook

This notebook enables you to import the definitions of DB2 tables into the TeamConnection database or to preview the results of this process by performing a trial run. The tables can be DB2 UDB tables or DB2/390 tables.

When you populate tables, related objects that are not already in the TeamConnection database are created.

The report displayed when a DB2 table has been populated is stored in a file named RDB *nnnnn* .RPT, where *nnnnn* is a generated hexadecimal number. The file is stored the directory specified on the General page of the Profile notebook.

Pages

- DB2
- TeamConnection

Push Buttons

- Populate
- Trial run
- Cancel
- Help

DB2 Page

This page enables you to:

⇑ Store the definitions of DB2 tables in the TeamConnection database

⇑ Qualify the names of the table definitions being stored

⇑ Associate table definitions with a DB2 system

⇑ Associate DB2/390 table definitions with a physical design

⇑ Perform a trial run to preview the results of this process

Lists

Tables Selected
Name

Fields

Variation
Revision
System
Relational Design
Physical Design
Use mapping table

Check Boxes

Reconcile
Use or create data elements for all columns

Push Button

Search
Clear

Tables Selected

This list shows the names of the tables you have selected to populate. Each entry is paired with an entry in the **Name** list.

Name

This list shows the default access names that correspond to objects in the **Tables selected** list. The names of related objects that are also populated are not shown.

System

This field contains the name you have chosen for grouping all the tables and associated objects that will be populated into the Team Connection database from a given DB2 UDB or DB2/390 catalog.

To fill in this field, you must either:

⇑ Search for and select a DB2 system object in the TeamConnection database or

⇑ Specify a DB2 system object in your profile, from which this field is initialized

In either case, you have to begin by creating the DB2 system object in the TeamConnection database. You can do this from a workfolder by selecting the **Create object** action under the **Workfolder** menu-bar choice.

Required?: Yes.

Relational Design

This field names an existing relational design, an object that identifies the tables you want to treat as a unit when using DataAtlas Modeler. (You must specify a relational design when you transform your tables from the relational definitions used by DB2 to the entity-relationship model used by DataAtlas Modeler.)

To fill in this field, you must either:

⇑ Search for and select a DB2 relational design object in the TeamConnection database or

⇑ Specify a DB2 relational design object in your profile, from which this field is initialized

Required?: Only if the populated tables will be used with DataAtlas Modeler.

Note: If you specify a relational design, DataAtlas creates a shareable data element for each column in each table that is populated. If the same column name exists in different tables, the revision field is incremented to create a unique shareable data element name.

Physical Design

This field appears only if you are populating DB2/390 tables. It names an existing physical design, an object that identifies the tables, indexes, table spaces, storage groups, and databases that you want to treat as a unit when using DataAtlas Designer.

To fill in this field, you must either:

⇑ Search for and select a DB2 physical design object in the TeamConnection database or

⇑ Specify a DB2 physical design object in your profile, from which this field is initialized

Required?: Only if the populated tables will be used with DataAtlas Designer.

Use Mapping Table

In this field specify the file that contains the mapping table you want to use for a reconcile operation. You can click **Search** to find the file you want; when you select the file, the field is filled in.

Reconcile

Check this box if you want DataAtlas to define populated columns in terms of the shareable data elements (SDEs) identified in a mapping table. You can specify the file that contains the mapping table in the **Use mapping table** field.

If the SDEs in the mapping table don't exist in the TeamConnection database, DataAtlas will create them.

Use or Create Data Elements for All Columns

Check this box if you want existing shareable data elements (SDEs) with the same name, data type, and length as the columns you will populate to be connected to those columns. If no such SDE exists for a given column, DataAtlas will create one.

Clear

Click this button to remove the name of the object from the adjacent field.

DB2 Subsystem - Search Window

This window enables you to select DB2 tables you want to populate.

Fields

- Database Alias
- Creator
- Table

Check Box

- Show matching system tables

Push Buttons

- Refresh (database alias list)
- Refresh (table list)
- Select all
- Deselect all
- Continue
- Cancel
- Help

Database Alias

Enter a wildcard pattern in this field. The database aliases that correspond to the pattern will be listed below when you click the adjacent **Refresh** button.

Required?: Yes.

Default: ' **%** '. Use only this to retrieve all database aliases.

Creator

Enter a wildcard pattern in this field. The tables whose creators correspond to this pattern, and whose names correspond to the wildcard pattern in the **Table** field, will be listed below when you click the adjacent **Refresh** button.

Required?: Yes.

Default: ' **%** '. Use only this to retrieve all creators.

Table

Enter a wildcard pattern in this field. The tables whose names correspond to this pattern, and whose creators correspond to the wildcard pattern in the **Creator** field, will be listed below when you click the adjacent **Refresh** button.

Required?: Yes.

Default: ' **%** '. Use only this to retrieve all tables.

Show Matching System Tables

Check this box if you want tables whose creator is SYSIBM to be included in the table list. You must click **Refresh** after checking the box to include the system tables in the list.

Refresh (Database Alias List)

Click this button to produce a list of database aliases that correspond to the wildcard pattern in the **Database alias** field.

Refresh (Table List)

Click this button to produce a list of tables that correspond to the wildcard patterns in the **Creator** and **Table** fields.

Select All

Click this button to select all the tables in the table list.

Deselect All

Click this button to deselect any tables you selected in the table list.

Continue

Click this button to open a populate notebook and populate the tables you selected in the table list.

DB2/390 Alias or Synonym Notebook

Use this notebook to create and maintain an alias or synonym object for a DB2/390 object.

Pages

- Definition
- Designs
- General

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

Menu-Bar Choices

- Alias or Synonym (Object management)
- Window
- Help

Definition Page

Use this page to specify a synonym or an alias for a table object or a view object.

Fields

⇑ Synonym
⇑ Alias
- Comment

  This field gives you room for your own notes and comments on the alias object.
- Label

⇑ Synonym (or Alias) for
- Table
- View

Push Buttons

Select **Search** to get a list of table or view objects that can be entered into the corresponding field. If you select an object from the list, it will appear in the field.

Select **Clear** to clear the corresponding entry field so that you can search for another object.

Task

Creating an alias or synonym object

Designs Page

On this page, you can assign the alias or synonym to a physical design.

Fields

In the **Physical designs** list box, you see the physical designs to which the alias or synonym object currently belongs.

Push Buttons

Select **Open** to open the selected physical design.

Select **Search** to search for an appropriate physical design among the list of existing physical designs, when you want to assign the alias or synonym to a physical design.

Select **Remove** to remove the alias or synonym object from the selected physical designs. The selected physical designs are deleted from the **Physical designs** list.

General Page

Use this page to specify general information about a DB2 alias or synonym object.

Once you have stored the object in the TeamConnection database, you cannot change the relational database qualifiers. The other specifications can be changed at any time.

Fields

⇑ Relational Database Qualifiers:
- System
- Creator

Push Buttons

Select **Search** to get a list of creators. When you select one, the **System** field is filled in along with the **Creator** field.

⇑ Name
⇑ Variation
⇑ Revision
⇑ Component

Push Button

Select **Search** to see a list of components represented in the TeamConnection database. When you select a component name from the list, it is put in the Component field.

⇑ Description
⇑ History

Name

This field contains the name of the DB2 alias or synonym object.

The value for the name must be a valid DB2 identifier .

Required?: Yes.

DB2/390 Column Notebook

Use the column notebook to design the selected column object. The notebook pages indicate the different design areas.

You can get design support for a column from anywhere in the notebook.

Pages

- Definition
- Options
- Data Load
- Work Load
- DB2 Actuals

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

Menu-Bar Choices

- Column (Object management)
- Designer
- Window
- Help

Tasks

- Designing a column object

Designer (for Column)

This menu-bar choice displays a pull-down menu with the following option:

- **Inform - Check completeness.** Get information about potential design problems in the column.

For more information, see Design support for the column object .

Definition Page

This page provides the information by which the column object is defined.

Fields

⇑ Name
⇑ Type
- Precision
- Scale
- Length
- Byte Count

⇑ Shareable data element

If you want to use a predefined shareable data element, select **Search** to search for the appropriate one. If you entered values in any of the data definition fields (**Type** , **Precision** , and so on), they will restrict the search to shareable data elements with the same values.

Select **Open** to open the notebook of the shareable data element.

Select **Remove** to clear the entry field and to disassociate the shareable data element from the column. Values in the data definition fields are also cleared.

⇑ Subtype:.
-Bit
-Mixed
-SBCS
-None

⇑ Description

Tasks

⇑ Defining a Column

Name

The value identifies the column name. The name must be unique within a table.

The value for the column name must be a valid DB2 long identifier .

Type

Available data types are:

INTEGER	This data type specifies a large integer.
SMALLINT	This data type specifies a small integer.
FLOAT	The data type specifies a floating-point number. The precision specifies the maximum number of digits. The precision must be a value from 1 to 53. 1 to 21 specifies a single precision floating point number. 22 to 53 specifies a double precision floating point number.
DECIMAL	The data type specifies a decimal number. The position of the decimal point is determined by the precision and the scale of the number. The precision defines the total number of digits. The scale defines the number of digits to the right of the decimal point. The precision can range from 1 to 31. The scale ranges from 0 up to the specified precision. If no precision is specified the default value 5 is used. If no scale is specified the default value 0 is used.
CHARACTER	This data type specifies a fixed length character string. The length can range from 1 to 254. If the length value is omitted, the default value 1 is used.
VARCHAR	This data type specifies a varying-length character string. The value can range from 1 to the maximum row size, if the column is the only column in the table. Otherwise, the sum of all byte counts must not exceed the maximum row size. A length value greater than 254 specifies a long string column.
LONG VARCHAR	This data type specifies a varying-length character string whose length is determined by the amount of space available in the page. The length value displays the character count for the field. It cannot be changed.
GRAPHIC	This data type specifies a fixed length graphic string. A graphic string can hold data from a double-byte character set. The length can range from 1 to 127. If the length value is omitted, the default value 1 is used.
VARGRAPHIC	This data type specifies a varying-length graphic string. The length can range from 1 to n/2 where n is the maximum row size. If the length value is omitted, the default value 1 is used. A length value greater than 127 specifies a long string column.
LONG VARGRAPHIC	This data type specifies a varying-length graphic string whose length is determined by the amount of space available in the page. The length value displays the character count for the field. It cannot be changed. A length value greater than 127 specifies a long string column.
DATE	This data type specifies a date. A date is a three part value (year, month, day).
TIME	This data type specifies a time. A time is a three part value (hour, minute, second).
TIMESTAMP	This data type specifies a time stamp. A time stamp is a seven part value (year, month, day, hour, minute, second, microsecond).
UNDEFINED	This data type is used in place of a DB2 data type that is not yet supported by DataAtlas. Currently, all DB2 data types are supported. If you generate DDL that specifies this data type, you have to modify the DDL to specify the data type the database requires.

Note: For the numeric limits of the various data types see the chapter *DB2 Limits* in the *DB2 SQL Reference*.

Precision

The application of the precision value is dependent on a column's data type. See the definition of a given data type for whether the precision value applies, and, if it does, which value it can take.

The precision value refines the column's data type if the data type is a floating-point number (FLOAT) or a decimal number (DECIMAL).

Scale

The scale value refines the column's data type if the data type is a decimal number (DECIMAL) . The scale value only applies to this data type.

See the definition of this data type for the type of values it can take.

Length

The application of the length value is dependent on the type of data type. See the definition of a given data type for whether the length value applies, and, if it does, which value it can take.

Byte Count

The byte count depends on the column's length value and its data type. It includes the DB2 internal overhead: It includes 1 byte overhead for each column which allows null values, 2 bytes overhead for varying length columns.

The byte count is the base for a table's row size. The maximum row size of a table depends on the page size of the assigned buffer pool and on whether EditProc has been specified.

EDITPROC	4K page size	32K page size
Y	4056	32714
N	4056	32704

It is a read-only field. The value is inserted automatically and cannot be changed.

Subtype

This value identifies the type of character data. The subtype can only be specified for the data types CHAR, VARCHAR, or LONG VARCHAR. The available options depend on the 'MIXED DATA' installation option.

None	This option specifies that the default option will be used. The default is SBCS, if the 'MIXED DATA' installation option is set to 'NO'. It is MIXED if the 'MIXED DATA' installation option is set to 'YES'.
SBCS	This option specifies that the column only holds data from a single-byte character set.
MIXED	This option specifies that the column holds data from a single-byte and double-byte character set. This option is only available if the 'MIXED DATA' installation option is set to 'YES'.
BIT	This option specifies that the column holds BIT data.

Shareable Data Element (SDE)

A shareable data element can be used to identify the data type of the column. When you use an SDE, you have a consistent definition of a data type everywhere this data type is used. When the data type changes, you need to change it only in the notebook of the shareable data element, but not with every occurrence.

If you want to use a predefined shareable data element, select **Search** to search for the appropriate one. If you entered values in any of the data definition fields (**Type** , **Precision** , and so on), they will restrict the search to shareable data elements with the same values.

Select **Open** to open the notebook of the shareable data element.

Select **Remove** to clear the entry field and to disassociate the shareable data element from the column. Values in the data definition fields are also cleared.

Options Page

Use this page to specify additional parameters for the selected column object.

Fields

⇑ Field Procedure :
-Name
-Parameters

⇑ Comment

The **Comment** field gives you room for your own notes and comments.

Check Boxes

NULLs not allowed
DEFAULT

Field Procedure

Here you specify the program name of the field procedure exit routine (FieldProc) for the column.

Name	Names a field procedure exit routine. The value must be a valid DB2 short identifier.
Parameters	The values are constants and serve as input parameters for the field procedure. They must be separated by a comma.

After the table has been implemented, the FieldProc for the column cannot be changed, added, or deleted.

FieldProc is optional. If you omit it, the column has no field procedure.

Note: For writing a field procedure exit routine see Appendix C (Volume 3) of the *IBM DB2 Administration Guide*.

NULLs Not Allowed

Check this box to prevent the column from containing null values.

Default: Unchecked.

DEFAULT

Check this box to specify a default value for the column if no value has been inserted. Select one of the radio buttons to specify the default value. (DB2/390 Version 4 or higher.)

Default: Unchecked.

The meanings of the buttons are as follows:

By data type	The default depends on the data type of the column.
Constant	The radio button enables the adjacent field, which specifies a default value for the column. The value must conform to the rules for assigning a value to a column.
User	The default is the execution-time value of the USER special register. You can specify this value only for a character string column with a length attribute of at least 8 bytes.
Current SQLID	The default is the SQL authorization ID (SQLID) of the process. You can specify this value only for a character string column with a length attribute of at least 8 bytes.
NULL	The default is NULL. This choice is not available if the **NULLs not allowed** box is checked on this page.

Data Load Page

Use this page to specify the data load for the selected column.

Fields

The following types of data load for column need to be specified:

- Number of distinct values
- Average length (if variable)

Number of Distinct Values

This integer value denotes the initial number of distinct values of the column. It must be less than or equal to the initial number of rows specified on the table data load page.

Average length

This integer value denotes the average length for variable length columns.

This value is used to calculate the required space for the table.

Work Load Page

Use this page to specify the work load for the selected column.

The column work load is specified in terms of frequency values. These frequency values are relative frequencies that specify how often an operation is applied to the column. The operations and their frequency of application are divided into update frequencies and usage frequencies.

Valid values for frequency range from 0 - 10, representing the scale low - medium - high.

The frequency values are used for the calculation of index proposals.

Fields

Frequencies:

- Update frequency
- Usage frequency in:
  - -Join operations
  - -WHERE clauses
  - -Range predicates
  - -Sorting functions

Update Frequency

This integer value denotes the expected update frequency for the column.

Usage Frequency

The frequency values of an operation specify how often this operation is applied to the column.

JOIN columns	The corresponding frequency value specifies how often the column is used in a join operation.
WHERE clause	The corresponding frequency value specifies how often the column is used in a WHERE clause.
Range predicates	The corresponding frequency value specifies how often the column is used in a 'Range' predicate.
	Range predicates are : &lt., >, &lt.=, >=, LIKE, BETWEEN
Sorting functions	The corresponding frequency value specifies how often the column is used in 'Sorting' functions.
	Sorting functions are: GROUP BY, ORDER BY, DISTINCT.

DB2 Actuals Page

On this page, you see the most important statistic values of the column from the DB2 catalog.

Fields

⇑ Number of distinct values
⇑ Second highest value
⇑ Second lowest value

Note:

Column values are extracted through extraction of the current values of the owning table. Use the **Extract statistics** function on the DB2 Actuals page of the Table notebook.

No data is available for extraction as long as the table object has not been implemented.

Number of Distinct Values

This integer value denotes the current number of distinct values of the column.

DB2 catalog name: SYSCOLUMNS.COLCARD.

Second Highest Value

This value denotes the second highest value of the column.

DB2 catalog name: SYSCOLUMNS.HIGH2KEY.

Second Lowest Value

This value denotes the second lowest value of the column.

DB2 catalog name: SYSCOLUMNS.LOW2KEY.

DB2/390 Database Notebook

Use the database notebook to maintain and design the selected database object. The notebook pages indicate the different design areas.

You can get design support for a database from anywhere in the notebook.

Pages

- Options
- Design Info
- Table Spaces
- Storage Group
- Buffer Pool
- Designs
- General

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

Menu-Bar Choices

- Database (Object management)
- Designer
- Window
- Help

Tasks

- Designing the database object

Designer (for Database)

This menu-bar choice displays a pull-down menu of design functions.

For the database object, you can get Information on potential design problems . Proposal and validation functions aren't available.

Options Page

Use this page to set the *read-only share* option and the default encoding scheme for data stored in the database.

Check Boxes

- ⇑ ROSHARE
- ⇑ CCSID

ROSHARE

DB2 allows you to read common DB2 data from a number of DB2 subsystems using shared read-only data. With shared read-only data, you can share the physical data for a given database among DB2 subsystems. However, one DB2 subsystem has exclusive control over updating the data in that shared database. The database in the system that can update is the *owner*. The same database in any other system is a *reader*.

Checking the **ROSHARE** box enables these buttons:

OWNER	If selected, the present database is the owner of the data in a shared read-only environment.
READ	If selected, the present database reads the data from another DB2 subsystem.

Design Info Page

The usage intent is necessary design information. Use this page to specify the usage intent for the database object. The usage intent of a database object can have the following values:

None	The usage intent is not specified. This is the default.
Other	The database is used for non QMF applications.
Query	The database is used for a QMF (Query Management Facility) application.

Note:

The usage intent of the database must match the usage intent of the assigned table spaces.

DataAtlas Designer checks this correspondence for you.

Table Spaces Page

On this page, you can view the table spaces assigned to the database. You see the currently assigned table spaces listed in the list box.

Storage Group Page

On this page, you can assign the database to a storage group. The current storage group is listed in the storage group field.

Push Buttons

Select **Open** to open the notebook of the storage group to which the database is assigned.

Select **Search** to search for an appropriate storage group among the list of existing storage groups.

Buffer Pool Page

Use this page to select a buffer pool for the database object.

The current buffer pool of the database is displayed. Further buffer pools are contained in the list box. Select one of them, if you want to use another default buffer pool for table spaces and indexes.

The buffer pool name on this page specifies the default buffer pool for the table spaces and indexes within the database. If no buffer pool has been specified for table space or index, the buffer pool specified here applies. The default is BP0.

32KB buffer pools apply only to table spaces. If a 32KB buffer pool is specified here, the default buffer pool for indexes in the database is BP0.

Designs Page

On this page, you can assign the database to a physical design.

Fields

In the **Physical designs** list box, you see the physical designs to which the database object currently belongs.

Push Buttons

Click **Inform** to get information on potential design problems.

Click **Open** to open the selected physical design.

Click **Search** to search for an appropriate physical design among the list of existing physical designs, when you want to assign the present object to a physical design.

Click **Remove** to remove the present design object from the selected physical design. The selected physical design is deleted from the list.

General Page

Use this page to specify general information about a DB2/390 database object.

Once you have stored the object in the TeamConnection database, you cannot change the relational database qualifiers. The other specifications can be changed at any time.

Fields

⇑ Relational Database Qualifiers:
- System
- Creator

Select **Search** to search for a list of items that can be entered into the corresponding field. If you select an item from the list, it will appear in the field.

⇑ Name
⇑ Variation
⇑ Revision
⇑ Component

Select **Search** to see a list of components represented in the TeamConnection database. When you select a component name from the list, it is put in the Component field.

⇑ Description
⇑ History

Name

This field contains the name of the DB2/390 database object.

The value for the name must be a valid DB2 short identifier . The name must be unique within the given location. It must not start with DSNDB, because these letters identify default names for databases.

Required?: Yes.

DB2/390 Index Notebook

Use the index notebook to maintain and design the selected index object. The notebook pages indicate the different design areas.

You can get design support for an index from anywhere in the notebook.

Pages

- Designs
- Definition
- Columns
- Storage
- Partitions
- DB2 Actuals
- Data Load
- General

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

Menu-Bar Choices

- Index (Object management)
- Designer
- Window
- Help

Tasks

- Designing an index object

Designer (for Index)

This menu-bar choice displays a pull-down menu with the types of design support for indexes:

- **Inform - Check completeness.** Get information about potential design problems in the index.
- **Validate - Check correctness.** Check the correctness of the index's design.
- **Propose - Suggest improvements.** Get suggested design improvements that you can execute.

Designs Page

On this page, you can assign the index to a physical design.

Fields

In the **Physical designs** list box, you see the physical designs to which the index object currently belongs.

Push Buttons

Select **Inform** to get information about potential design problems in the index.

Select **Validate** to check the correctness of the index's design.

Select **Propose** to get suggested design improvements that you can execute.

Select **Open** to open the selected physical design.

Select **Search** to search for an appropriate physical design among the list of existing physical designs, when you want to assign the index to a physical design.

Select **Remove** to remove the index object from the selected physical designs. The selected physical designs are deleted from the **Physical designs** list.

Definition Page

On this page, you can provide the information by which the index object is defined.

Fields

⇑ Owning table

Select **Open** to open the notebook of the owning table.

Select **Search** to search for the table on which you want to define the index.

⇑ Piece size

Check Boxes

⇑ Type
⇑ Unique
⇑ WHERE NOT NULL
⇑ Password
⇑ Subpages
⇑ Buffer pool
⇑ Defer
⇑ Close
⇑ Clustered
⇑ Partitioned
⇑ Number of partitions

Owning Table

The value identifies the table on which the index is defined. It must be a valid table name of an existing table.

Select **Open...** to open the notebook of the owning table.

Select **Search...** to search for the table on which you want to define the index.

Piece Size

In this field, available for DB2/390 Version 5 or higher, you can specify the maximum piece size for a nonpartitioned index. Use the adjacent drop-down list to specify the unit size - K (kilobytes), M (megabytes), or G (gigabytes) - of the piece size number.

Valid piece-size values are:

0K
258K
512K
1024K (or 1M)
2048K (or 2M)
4096K (or 4M)
8192K (or 8M)
16384K (or 16M)
32768K (or 32M)
65536K (or 64M)
131072K (or 128M)
282144K (or 256M)
524288K (or 512M)
1048576K (or 1024M or 1G)
2097152K (or 2048M or 2G)
4194304K (or 4096M or 4G)

Required?: No.

Default: 4G for indexes backed by large table spaces; otherwise, 2G.

Type

Check this box to enable the **1** and **2** radio buttons. Then choose a button to indicate that this is a Type 1 or Type 2 index. (Available for DB2/390 Version 4 or higher.)

Restrictions: Do not specify 1 if the table space containing the owning table has a lock size of ROW. If you specify 1, the only allowed value with the **Subpages** check box is 1. If you specify 2, the **Subpages** value is ignored.

Default: If you do not check this box, the default index type is determined as follows:

⇑ If the lock size is ROW, the index type is 2.

⇑ If the lock size is not ROW, the index type is the type specified in the DEFAULT INDEX TYPE field on the DSNTIPE installation panel.

Unique

A unique index does not permit duplicate values in index columns.

Insertion of a new row fails, if another row with the same ID value already exists in the table.

A unique index is required for primary key columns and unique key columns.

WHERE NOT NULL

This check box, available for DB2/390 Version 4 or higher, determines how null values in key columns are evaluated. If you don't check it, any two null values are taken to be equal. If the key is a single column, it can contain no more than one null value.

If you do check it, any two null values are taken to be unequal. If the key is a single column, it can contain any number of null values, although the other values must be unique. You can check this box only if this is a Type 2 index.

Password

This field specifies a master level password that is sent to access method services when the data sets of the index are used by DB2.

The value for the password is a valid DB2 short identifier .

If you use a storage group, this password protects the data sets as well as it is the one passed to access method services when the data sets are used by DB2. If you are defining the data sets yourself, the password is only the one that is passed on to VSAM (virtual storage access method); you need to define separately the password that protects the data sets.

All password-protected data sets of the index must have the same password.

Subpages

The page size on an index is fixed at 4K. Under the data portion a page can be 4K or 32K, and every time a row is locked in a data page, the entire page is locked. This can lead to slow response times. Therefore, DB2 permits a part of an index page (SUBPAGE) to be locked, instead of locking the entire 4K. The subpage is set when the index is created and can have one of the following values:

1	Entire page
2	Half a page
4	Quarter of a page
8	Eighth of a page
16	Sixteenth of a page

Buffer pool

This field identifies the buffer pool to be used for the index.

The buffer pool name must identify an activated 4KB buffer pool. No 32KB buffer pools can be used for indexes.

If this option is not set, the default buffer pool is used. The default buffer pool is the buffer pool of the database. If the default buffer pool of the database is a 32KB page buffer pool, the default is BP0.

Defer

This option allows you to build the index later, instead of directly with the creation of an index on a table. This is especially useful for large tables.

When you defer the build for DB2-managed data sets, DB2 creates the VSAM data set and adds the definition for the index to the catalog, but does not build the index. For user-managed data sets, the data must already exist before DB2 adds the definition to the catalog.

DEFER=NO	The index is built during creation. This is the default.
DEFER=YES	The index is not built during creation. The index is placed in recover pending status. It must be recovered by the RECOVER INDEX utility.

Close Option

Check this box if you want DB2 to close the data sets when they are no longer used. This is the default.

Clustered

This field specifies whether the index is clustered.

With a clustering index the rows are stored in physical order based on the values in the key.

Only one clustering index in a table is permitted.

If you do not specify any index as clustered, DB2 defines the first index created on the table as a clustered index.

Partitioned

The partitioned index is a special form of the clustered index, since each of the individual parts is clustered.

As with a clustered index, only one partitioned index can be used for a table.

Note: Index partitions and the table space partitions must match.

Number of Partitions

The number of index partitions is limited to 64.

If the corresponding table is contained in a partitioned table space and the table space has n partitions, the number of index partitions must range from 1 through n.

Columns Page

Use this page to specify the columns for the creation of an index.

Fields

The columns available for the creation of an index are listed in the **Available Columns** list box.

The columns selected for the creation of an index are listed in the **Selected Columns** list box.

For a selected index column, select one of the following **Sorting orders** :

⇑
⇑ Ascending

This option puts the column entries in ascending order (this is the default).
⇑ Descending

This option puts the column entries in descending order.

Push Buttons

Select **Add** to add the selected columns to the index.

Select **Remove** to remove the selected columns from the index.

Storage Page

Use this page to specify the necessary storage information. This includes specifying free space and whether the data sets for the index are to be managed by the user (VCAT option) or managed by DB2 (Storage Group option). You make these specifications by using the items within the **Using block** (for DB2/390 Version 4 or higher), **Free block** , and **GBPCACHE block** (for data sharing environments, DB2/390 Version 4 or higher). Checking the box beside a block enables the items within it.

If you omit the **Using block** information, the data sets are managed by DB2 on volumes listed in the default storage group of the table's database; primary and secondary quantity (PRIQTY, SECQTY) and the Erase check box assume their default values.

Check Box

⇑ **Compress** : Specifies whether data compression applies to the rows of the index. If the option is set, the rows are not compressed until the LOAD or REORG utility is run on the index.

Blocks

⇑ Using block
- **VCAT**
  - **Catalog name**
- **Storage group**
  - **Name**

    Select **Search** to search for a list of existing storage groups to select from.
  - **PRIQTY**
  - **SECQTY**

    Select **Convert** to convert primary and secondary quantity values from kilobytes to the number of cylinders or tracks.
  - **Erase**

⇑ Free block
- **FREEPAGE**
- **PCTFREE**

⇑ GBPCACHE
- **Changed**
- **All**

VCAT (Volume Catalog)

Selecting the VCAT option means that the data sets are managed by the user.

The data set name starts with the specified catalog name. The catalog name identifies the integrated catalog facility (ICF) catalog for the storage group.

The value for the catalog name is a short DB2 identifier . An alias must be used if the name is longer than 8 characters. The catalog name cannot be altered.

Storage Group

Selecting the storage group option means that DB2 will define and manage the data sets for the index or table space. Each data set will be defined on a volume listed in the identified storage group.

The values to define are:

⇑ Name

Select **Search** to search for a list of existing storage groups to select from.

⇑ PRIQTY

The integer value specifies the minimum primary space allocation for a DB2-managed data set.

⇑ SECQTY

The integer value specifies the minimum secondary space allocation for a DB2-managed data set.

⇑ Erase

Check this box if you want DB2-managed data sets to be overwritten with zeros before they are erased during the execution of a utility or when the index or table space is dropped.

Note: You can request a proposal from DataAtlas Designer for the PRIQTY and SECQTY fields and the Erase check box.

FREEPAGE

The FREEPAGE integer value specifies how often to leave a page of free space when index entries are created as the result of executing a DB2 utility or when creating an index for a table with existing rows.

The default is 0, leaving no free pages.

Note: For this value you can request a proposal from DataAtlas Designer.

PCTFREE

The PCTFREE integer value determines the percentage of free space to leave in each nonleaf page and subpage when entries are added to the index or index partition as the result of executing a DB2 utility or when creating an index for a table with existing rows.

Note: For this value you can request a proposal from DataAtlas Designer.

Changed

Select this button if you want updated pages to be written to the group buffer pool when:

⇑ More than one member in the data sharing group has the index or partition open and

⇑ At least one member has it open for update

All

Select this button if you want pages to be cached in the group buffer pool as they are read from DASD.

Exception: If only one DB2 subsystem is updating when no other DB2 subsystems have an interest in the page set, no pages are cached in the group buffer pool.

Partitions Page

Use this page to define and maintain each index partition.

To create an index partition, switch to the notebook of the owning table and request a proposal for creating indexes.

Note:

You can get a proposal for a partitioned index only if the table is partitioned.

Fields

Partition definition
The following defining information is listed for each partition:
⇑ Partition number
⇑ Partition column
⇑ Range of partition column (Upper value)

Partition
The following type and storage information is specified for each partition:
⇑ Partition

Shows the partition number.

⇑ Type

Indicates whether the data sets for the index are to be managed by the user (VCAT) or managed by DB2 (Storage group).

⇑ VCAT

Specifies the volume catalog name.

⇑ Storage group

Specifies the storage group name.

⇑ PRIQTY

Specifies the minimum primary space allocation for a DB2-managed data set.

⇑ SECQTY

Specifies the minimum secondary space allocation for a DB2-managed data set.

⇑ ERASE

Specifies whether DB2-managed data sets are to be erased when they are deleted during the execution of a utility or when the index is dropped.

⇑ FREEPAGE

Specifies how often a free page must be reserved in the index space.

⇑ PCTFREE

Specifies the percentage of space that must be left empty in each page.

⇑ GBPCACHE

Specifies what index pages are written to the group buffer pool in a data sharing environment. (DB2/390 Version 4 or higher.)

Select **Open** to open the DB2/390 Index Partitions window and specify or change the storage information for a selected partition.

Note: The number of index partitions must match the number of table space partitions, as well as the number of table partitions.

DB2 Actuals Page

On this page, you see the most important statistic values of the index or its partitions from the DB2 catalog, as extracted during the last extract run.

Fields

DB2 statistic values for the index (SYSIBM.SYSINDEXES):

- Percentage of rows in clustering order
- Number of distinct values of key
- Number of distinct values of first key column
- Number of leaf pages
- Number of levels in index tree
- Last RUNSTATS time
- Last RUNSTATS date
- Space (KB)
- Clustered by the index

DB2 statistic values for the index **Partitions** (SYSIBM.SYSINDEXPART):

- Referenced rows
- Far rows
- Near rows
- Leaf distance
- Space
- RUNSTATS time
- RUNSTATS date

Note:

The index values are extracted through extraction of the current values of the owning table. Use the **Extract statistics** function on the DB2 Actuals page of the table notebook.

No data is available for extraction as long as the table object has not been implemented.

Percentage of Rows in Clustering Order

This integer value shows the percentage of rows, multiplied by 100, that are in clustering order.

DB2 catalog name: SYSINDEXES.CLUSTERRATIO.

Number of Distinct Values of Key

This integer value denotes the number of distinct values in the entire key, that is in all of the index columns.

DB2 catalog name: SYSINDEXES. FULLKEYCARD.

Number of Distinct Values of First Key Column

This integer value denotes the number of distinct values in a single indexed column or in the leading column of a composite index (first key column).

DB2 catalog name: SYSINDEXES.FIRSTKEYCARD.

Number of Leaf Pages

This integer value denotes the number of entire pages contained in the leaves of an index tree.

DB2 catalog name: SYSINDEXES.NLEAF.

Number of Levels in Index Tree

This integer value denotes the number of levels in an index tree.

DB2 catalog name: SYSINDEXES.NLEVELS.

Space (KB)

This integer value denotes the number of kilobytes of   DASD storage allocated to the index, index partition, table space, or table space partition (input from STOSPACE utility).

DB2 catalog name:

SYSINDEXES.STATSTIME.

SYSINDEXPART.STATSTIME.

SYSTABLESPACE.SPACE.

SYSTABLEPART.SPACE.

Clustered by the Index

This value specifies whether the table is clustered by the index.

Yes	Most rows are in clustering order.
No	A significant number of rows are not in clustering order.

DB2 catalog name: SYSINDEXES.CLUSTERED.

Referenced Rows

This integer value denotes the number of rows referenced by the index or the index partition.

DB2 catalog name: SYSINDEXPART.CARD.

Far Rows

This integer value denotes the number of referenced rows far from an optimal position because of an insert into a full page.

DB2 catalog name: SYSINDEXPART.FAROFFPOS.

Near Rows

This integer value denotes the number of referenced rows in a near but not optimal position, because of an insert into a full page.

DB2 value: NEAROFFPOS.

DB2 catalog name: SYSINDEXPART.NEAROFFPOS.

Leaf Distance

This integer value denotes 100 times the average number of leaf pages between successive active leaf pages of the index.

DB2 catalog name: SYSINDEXPART.LEAFDIST.

Data Load Page

This page applies only to non-unique indexes. Use it to specify values that help DataAtlas Designer calculate space for a type 2 index.

Fields

- Number of distinct key values...
- Number of data records...

Number of Distinct Key Values...

In this field enter the number of distinct keys (count non-unique keys only once) in the index.

Number of Data Records...

In this field enter number of rows in the index.

General Page

Use this page to specify general information about a DB2/390 index object.

Once you have stored the object in the TeamConnection database, you cannot change the relational database qualifiers. The other specifications can be changed at any time.

Fields

⇑ Relational Database Qualifiers:
- System
- Creator

Select **Search** to search for a list of items that can be entered into the corresponding field. If you select an item from the list, it will appear in the field.

⇑ Name
⇑ Variation
⇑ Revision
⇑ Component

Select **Search** to see a list of components represented in the TeamConnection database. When you select a component name from the list, it is put in the Component field.

⇑ Description
⇑ History

Name

This field contains the name of the DB2/390 index object.

The value for the name must be a valid DB2 identifier .

Required?: Yes.

DB2/390 Index Partitions Window

Use this window to view or specify storage information for an index partition. This includes specifying free space and whether the data sets for the partition are to be managed by the user (VCAT option) or managed by DB2 (Storage Group option). You make these specifications by using the items within the **Using block** (for DB2/390 Version 4 or higher), **Free block**, and **GBPCACHE block** (for data sharing environments, DB2/390 Version 4 or higher). Checking the box beside a block enables the items within it.

If you omit the **Using block** information, the data sets are managed by DB2 on volumes listed in the default storage group of the table's database; primary and secondary quantity (PRIQTY, SECQTY) and the **Erase** check box assume their default values.

Check Box

⇑ **Compress** : Specifies whether data compression applies to the rows of the index partition. If the option is set, the rows are not compressed until the LOAD or REORG utility is run on the table in the index partition.

Blocks

⇑ Using block
- **VCAT**
  - **Catalog name**

- **Storage group**
  - Name

    Select **Search** to search for a list of existing storage groups to select from.
  - **PRIQTY**
  - **SECQTY**

    Select **Convert** to convert primary and secondary quantity values from kilobytes to the number of cylinders or tracks.
  - **Erase**

⇑ Free block
- **FREEPAGE**
- **PCTFREE**

⇑ GBPCACHE
- **Changed**
- **All**

DB2/390 Physical Design Notebook

This notebook enables you to:

⇑ Open the properties notebooks of the objects that belong to a DB2/390 physical design

⇑ Validate the objects, collect information about them, and execute design actions on them

The notebook' pages correspond to the objects that can be in a DB2/390 physical design - tables, table spaces, storage groups, indexes, views, and aliases or synonyms.

Menu-Bar Choices

Physical design
Window
Help

Pages

Alias/Synonym
Database
General
Index
Storage Group
Table
Table Space
View

Push Buttons

OK
Save
Cancel
Help

Physical Design

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

Alias/Synonym Page

This page enables you to view the list of aliases and synonyms that belong to the physical design and open the properties notebook for an entry selected from the list.

List

Aliases/Synonyms

Push Button

Open
Search
Remove

Aliases/Synonyms

This list identifies the aliases and synonyms that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook.

Open

Click this button to open the properties notebook for an entry you selected in the list.

Search

Click this button to find an alias/synonym object in the TeamConnection database and add it to the **Aliases/Synonyms** list. Adding an alias/synonym to the list associates it with the physical design.

Remove

Click this button to remove the entries you selected from the **Aliases/Synonyms** list and from this physical design. The alias/synonym objects will still exist in the TeamConnection database.

Database Page

This page enables you to:

⇑ View the list of databases that belong to the physical design
⇑ Open the properties notebook for an entry selected from the list
⇑ Get information about the entry

List

Databases

Push Buttons

Open
Search
Remove
Inform

Databases

This list identifies the databases that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook. You can use the **Inform** push button to collect information about the entry.

Search

Click this button to find a database object in the TeamConnection database and add it to the **Databases** list. Adding a database to the list associates it with the physical design.

Remove

Click this button to remove the entries you selected from the **Databases** list and from this physical design. The database objects will still exist in the TeamConnection database.

Inform

Use this push button to see a menu of actions that can produce a report with information about the selected entry.

General Page

Use this page to specify general information about a DB2/390 physical design.

Fields

- Relational design
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Relational Design

If you opened this notebook by using the **Create** push button in a Relational Design notebook , this field will be filled in with the name of that relational design.

If you are choosing a relational design to associate with this physical design, click the adjacent **Search** button to see a list you can choose from.

The variation and revision names that belong to the relational design will appear in the **Variation** and **Revision** fields.

Required?: Yes.

Name

This field contains the access name of the physical design.

Required?: Yes.

Index Page

This page enables you to:

⇑ View the list of indexes that belong to the physical design
⇑ Open the properties notebook for an entry selected from the list
⇑ Take design actions that relate to the entry

List

Indexes

Push Buttons

Open
Search
Remove
Propose
Validate
Inform

Indexes

This list identifies the indexes that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook. You can use the **Propose** , **Validate** , and **Inform** push buttons to execute design actions on the selected entry, identify possible errors or inconsistencies in it, and collect information about it.

Search

Click this button to find an index object in the TeamConnection database and add it to the **Indexes** list. Adding an index to the list associates it with the physical design.

Remove

Click this button to remove the entries you selected from the **Indexes** list and from this physical design. The index objects will still exist in the TeamConnection database.

Propose

Use this push button to see a menu of design actions that relate to the selected entry. If you can request a report on an action and the report recommends it, you can execute the action from the report.

Validate

Use this push button to see a menu of actions that can produce a report identifying errors or inconsistencies in the selected entry.

Storage Group Page

This page enables you to:

⇑ View the list of storage groups that belong to the physical design
⇑ Open the properties notebook for an entry selected from the list
⇑ Take design actions that relate to the entry

List

Storage groups

Push Buttons

Open
Search
Remove
Propose
Validate
Inform

Storage Groups

This list identifies the storage groups that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook. You can use the **Propose** , **Validate** , and **Inform** push buttons to execute design actions on the selected entry, identify possible errors or inconsistencies in it, and collect information about it.

Search

Click this button to find a storage group object in the TeamConnection database and add it to the **Storage groups** list. Adding a storage group to the list associates it with the physical design.

Remove

Click this button to remove the entries you selected from the **Storage groups** list and from this physical design. The storage group objects will still exist in the TeamConnection database.

Table Page

This page enables you to:

⇑ View the list of tables that belong to the physical design
⇑ Open the properties notebook for an entry selected from the list
⇑ Take design actions that relate to the entry

List

Tables

Push Buttons

Open
Search
Remove
Propose
Validate
Inform

Tables

This list identifies the tables that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook. You can use the **Propose** , **Validate** , and **Inform** push buttons to execute design actions on the selected entry, identify possible errors or inconsistencies in it, and collect information about it.

Search

Click this button to find a table object in the TeamConnection database and add it to the **Tables** list. Adding a table to the list associates it with the physical design.

Remove

Click this button to remove the entries you selected from the **Tables** list and from this physical design. The table objects will still exist in the TeamConnection database.

Table Space Page

This page enables you to:

⇑ View the list of table spaces that belong to the physical design
⇑ Open the properties notebook for an entry selected from the list
⇑ Take design actions that relate to the entry

List

Table spaces

Push Buttons

Open
Search
Remove
Propose
Validate
Inform

Table spaces

This list identifies the table spaces that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook. You can use the **Propose**, **Validate**, and **Inform** push buttons to execute design actions on the selected entry, identify possible errors or inconsistencies in it, and collect information about it.

Search

Click this button to find a table space object in the TeamConnection database and add it to the **Table Spaces** list. Adding a table space to the list associates it with the physical design.

Remove

Click this button to remove the entries you selected from the **Table Spaces** list and from this physical design. The table space objects will still exist in the TeamConnection database.

View Page

This page displays the list of views that belong to the physical design. You can open the properties notebook for an entry selected from the list.

List

- Views

Push Button

- Open
- Search
- Remove

Views

This list identifies the views that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook.

Search

Click this button to find a view object in the TeamConnection database and add it to the **Views** list. Adding a view to the list associates it with the physical design.

Remove

Click this button to remove the entries you selected from the **Views** list and from this physical design. The view objects will still exist in the TeamConnection database.

DB2/390 Storage Group Notebook

Use the storage group notebook to maintain and design the selected storage group object. The notebook pages indicate the different design areas.

You can get design support for a storage group from anywhere in the notebook.

Pages

- Designs
- Storage
- Databases
- Table Spaces
- Indexes
- Design Info
- DB2 Actuals
- General

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

Menu-Bar Choices

- Storage Group (Object management)
- Designer
- Window
- Help

Tasks

- Designing a storage group object

Designer (for Storage Group)

This menu-bar choice displays a pull-down menu with the types of design support for storage groups :

⇑ **Inform - Check completeness.** Get information about potential design problems in the storage group.

⇑ **Validate - Check correctness.** Check the correctness of the storage group's design.

⇑ **Propose - Suggest improvements.** Get suggested design improvements that you can execute.

Designs Page

On this page, you can assign the storage group to a physical design.

Fields

In the **Physical designs** list box, you see the physical designs to which the storage group object currently belongs.

Push Buttons

Select **Inform** to get information about potential design problems in the storage group.

Select **Validate** to check the correctness of the storage group's design.

Select **Propose** to get suggested design improvements that you can execute.

Select **Open** to open the selected physical design.

Select **Search** to search for an appropriate physical design among the list of existing physical designs, when you want to assign the storage group to a physical design.

Select **Remove** to remove the storage group from the selected physical design. The selected physical design is deleted from the list.

Storage Page

Use this page to specify a volume catalog and the kind of device type needed, and to choose between storage management by the system or your own storage management. If you decide for your own storage management, you also allocate the volumes for the storage group here.

Fields

⇑ VCAT
- Catalog name
- Password protection

⇑ Device type
⇑ Storage management by the system (SMS)
⇑ Storage management by the user
-Volumes

Volumes are identified by their volume serial number (VOLSER).

Available volumes are listed in the **Available** list box. The currently specified volumes are listed in the **Selected** list box.

Select **Add** to add selected volumes to the list of volumes selected for allocation.

Select **Remove** to remove selected volumes from the list of volumes selected for allocation.

Select **Add volume** to add the volume specified in the entry field to the list of available volumes.

Select **Delete volume** to delete the volume specified in the entry field from the list of available volumes.

VCAT (Volume Catalog)

The volume catalog name identifies the VSAM integrated catalog facility (ICF) catalog for the storage group.

The value for the catalog name is a short DB2 identifier . An alias must be used if the name is longer than 8 characters. The catalog name cannot be altered.

Password Protection of Catalogs

If the ICF catalog is password protected, the master ICF catalog password must be specified in the generated DLL statement.

Option set	Password required
Option not set	Password not required

Device Type of Volumes

This field specifies the device type of the volumes.

There are slow or fast device types.

The field is optional, and can be used for information purposes only.

SMS (Storage Management System)

Storage management by the system (SMS) is indicated by an asterisk for the VOLUMES attribute.

See the *DB2 Administration Guide* for a detailed description of using SMS to manage data sets.

Volume Identifier (VOLSER)

Each volume-identifier is a volume serial number with a maximum of 6 characters. The volume-identifiers must be of the same device type.

The maximum number of volumes is 133. The same volume-ID must not be specified twice.

The volume identifiers are required for design implementation.

Note: DataAtlas Designer does not check whether volume-identifiers are in fact of the same device type.

Databases Page

On this page, you can assign databases to the storage group.

<u>Fields</u>

You see the currently assigned databases in the list box.

<u>Push Button</u>

Select **Open** to open the notebook of a selected database.

Table Spaces Page

This page shows the table spaces assigned to the storage group. The table spaces currently assigned are listed in the list box.

You cannot assign any table spaces to the storage group on this page. To assign a table space to the storage group, you need to open the notebook of the table space.

Push Buttons

Select **Open** to open the notebook of a selected table space.

Indexes Page

This page shows the indexes assigned to the storage group. The indexes currently assigned are listed in the list box.

You cannot assign any indexes to the storage group on this page. To assign an index to the storage group, you need to open the notebook of the index.

Push Buttons

Select **Open** to open the notebook of a selected index.

Design Info Page

On this page, you specify the usage intent of the storage group and the space required for the storage group.

Fields

The **usage intent** of a storage group can be one of the following:

Table spaces (general)	The storage group is reserved for the storage of an arbitrary number of table spaces.
Indexes (general)	The storage group is reserved for the storage of an arbitrary number of indexes.
Table space	The storage group is reserved for the storage of one specific table space.
	Select **Search** to search for the appropriate table space.
Index	The storage group is reserved for the storage of one specific index.
	Select **Search** to search for the appropriate index.
None	The storage group is not reserved for any specific type of data structure. Both, table spaces and indexes can be assigned to the storage group.

The integer value for **required space** determines the allocation of the appropriate volumes.

The value must be entered in kilobytes. It can also be displayed in terms of cylinders or tracks. Select **Convert** to convert the required space value from kilobytes to the number of cylinders or tracks.

Note: On request, DataAtlas Designer calculates the required space value for you and offers a proposal.

For an optimum proposal you need to have assigned the necessary table spaces and created the necessary indexes. Also, the following table data load specifications are needed:

⇑ Initial number of rows
⇑ Confidence factor

DB2 Actuals Page

On this page, you see the most important statistic values of the storage group from the DB2 catalog.

Fields

DB2 statistic value:

⇑ Used space (SYSSTOGROUP.SPACE)

The amount of used space is given in terms of kilobytes.

Select **Convert** to convert the used space value from kilobytes to the number of cylinders or tracks.

Note:

The storage group values are extracted through extraction of the values of one of the table spaces assigned to the storage group. Use the **Extract statistics** function on the DB2 Actuals page of the table space notebook.

If the table spaces have not been implemented yet, no extraction data is available.

General Page

Use this page to specify general information about a DB2/390 storage group object.

Once you have stored the object in the TeamConnection database, you cannot change the relational database qualifiers. The other specifications can be changed at any time.

Fields

⇑ Relational Database Qualifiers:
- System
- Creator

Select **Search** to search for a list of items that can be entered into the corresponding field. If you select an item from the list, it will appear in the field.

⇑ Name
⇑ Variation
⇑ Revision
⇑ Component

Select **Search** to see a list of components represented in the TeamConnection database. When you select a component name from the list, it is put in the Component field.

⇑ Description
⇑ History

Name

This field contains the name of the DB2/390 storage group object.

The value for the storage group name must be a valid DB2 short identifier .

If you want to create a new storage group, the name must be unique and not match an existing one. Changing the name of an existing storage group requires to drop the storage group and to create a new one.

Required?: Yes.

DB2/390 Table Notebook

Use the table notebook to maintain and design the selected table object. The notebook pages indicate the different design areas.

You can get design support for a table from anywhere in the notebook.

Pages

- Definition
- Options
- Primary Key
- Unique Keys
- Foreign Keys
- Checks
- Indexes
- Table Space
- Partitions
- Data Load
- Work Load
- DB2 Actuals
- Designs
- General

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

Menu-Bar Choices

- Table (Object management)
- Designer
- Window
- Help

Tasks

- Designing a table object

Reset

Click this button to restore your last saved values.

DB2 Identifiers

Identifiers are tokens used to form names.

There are two kinds of identifiers:

short identifiers (8 characters) and long identifiers (18 characters).

Note: For detailed information on valid characters see the 'DB2 SQL Reference'.

Object Management

For each object a set of general actions of object management is available.

Pull-down summary

Save

Save

Select this pull-down choice to save the settings in this notebook and keep it open.

Recommendation: Use this pull-down choice intermittently to save your work when you are in a long updating session.

Designer (for Table)

This menu-bar choice displays a pull-down menu with the types of design support for tables :

⇑ **Inform - Check completeness.** Get information about potential design problems in the table.

⇑ **Validate - Check correctness.** Check the correctness of the table's design.

⇑ **Propose - Suggest improvements.** Get suggested design improvements that you can execute.

Definition Page

On this page, you see the table definition on which the selected DB2/390 table is based, and a table listing the table's columns and their characteristics.

Fields

Table Definition

Columns

Tasks

- ⇑ Viewing the shareable table definition
- ⇑ Viewing the columns
- ⇑ Adding and deleting columns
- ⇑ Optimizing the table layout

Table Definition

The current table definition is listed in the **Table definition** entry field.

Select **Open** to open the notebook of the listed shareable table definition.

Note: On the Definition page of the **Shareable Table Definition** notebook, you can assign this table definition to a relational design. You have to take this action to use DataAtlas Modeler with the table definition.

Columns

The information on the table's columns is represented in a table:

⇑ Name
⇑ Type
⇑ Length
⇑ Precision
⇑ Scale
⇑ Subtype
⇑ Byte count
⇑ Data Element
⇑ Comment

Push Buttons

Select **Create...** to create a new column for the table object.

An empty column notebook displays. When the notebook is closed, the specified information is listed in the table with the table's columns.

Select **Open...** to change information of the selected column.

The notebook of the selected column displays. When the notebook is closed, the specified information is listed in the table with the table's columns.

Select **Delete** to delete the selected columns.

Select **Move up** to change the table layout by moving a selected column up in the table.

Select **Move down** to change the table layout moving a selected column down in the table.

Options Page

Use this page to specify routines and other options for the selected table object.

Fields

Procedures
⇑ EditProc
⇑ ValidProc

OBID
Label
Comment

Check Boxes and Radio Buttons

CCSID
⇑ ASCII
⇑ EBCDIC

Restrict on drop
Audit
⇑ None
⇑ Changes
⇑ All

Data capture
⇑ None
⇑ Changes

Procedures

Here you specify the following procedures:

EditProc	The value specifies an edit routine for the table.
	It must a be valid DB2 short identifier . After the table has been implemented, an edit procedure cannot be added, changed, or deleted.
ValidProc	The value specifies a program name as a validation exit routine for the table.
	It must a be valid DB2 short identifier .

If you omit EDITPROC or VALIDPROC, the table has no edit procedure or validation procedure.

Note: For writing an edit routine or a validation exit routine see Appendix C (Volume 3) of the *IBM DB2 Administration Guide* .

OBID

An OBID is an integer value which identifies the object's internal descriptor. This value is required if the database for the table was defined as ROSHARE READ.

Label

Like the name of an object, the label identifies the object.

But the label is optional, while the name is mandatory.

Comment

Use this field for notes or remarks about the physical object you are defining.

CCSID

The letters stand for "coded character set ID." If you check the box, the **ASCII** and **EBCDIC** radio buttons are enabled. You can use them to specify the default encoding scheme for data stored in the database. Supported for DB2/390 Version 5 or higher.

ASCII	Select this button to specify that data must be encoded in ASCII.
EBCDIC	Select this button to specify that data must be encoded in EBCDIC.

Restrict on Drop

Check this box to specify that the table cannot be dropped. A check also prevents the database and table space containing the table from being dropped. Supported for DB2 Version 4 and higher.

Audit

The audit option identifies the type of access to this table that causes auditing to be performed. If no value is specified, the default 'None' is used.

None	Specifies that no auditing is to be done when this table is accessed. This is the default.
Changes	Specifies that auditing is to be done when the table is accessed during the first insert, update, or delete operation performed by each unit of work. However, the auditing is done only if the appropriate audit trace class is active.
All	Specifies that auditing is to be done when the table is accessed during the first operation of any kind performed by each unit of work of a utility or application process. However, the auditing is done only if the appropriate audit trace class is active and the access is not performed with COPY, RECOVER, REPAIR, or any stand-alone utility.

Data Capture

The data capture option specifies whether the logging of SQL INSERT, UPDATE, and DELETE operations on the table is augmented by additional information.

None	Do not record additional information in the log. This is the default.
Changes	Write additional data about SQL updates to the log.

Primary Key Page

This page contains information on the table's primary key.

The primary key clearly identifies the rows of a table. It can consist of one or more columns. Only unique values are allowed in primary key columns.

Fields

The currently specified primary key columns are listed in the **Selected** list box. All of the columns available for the creation of the table's primary key are listed in the **Available** list box.

Push Buttons

Select **Add** to add the selected columns to the primary key.

Select **Remove** to remove the selected columns from the primary key.

Tasks

⇑ Creating and modifying a primary key

Unique Keys Page

On this page, you can specify columns for the composition of unique keys.

Fields

The currently specified unique keys are listed in the **Unique keys** list.

All of the columns available for the creation of unique keys are listed in the **Available** list box. The columns selected for the creation of a unique key are listed in the **Selected** list box.

Push Buttons

To create a unique key, click &lt.NEW>, select from available columns for the unique key, and click **Create** .

To modify a unique key, select a key in the **Unique keys** list, modify its selected columns, and click **Modify** .

To delete a unique key, select a key in the **Unique keys** list and click **Delete** .

To add selected columns to the set of columns that should compose a unique key, click **Add** .

To remove selected columns from the set of columns that should compose a unique key, click **Remove** .

Tasks

⇑ Creating and modifying unique keys

Foreign Keys Page

On this page, you can create foreign keys (constraints) and specify the referential constraint (delete rule) over these keys.

Fields

- Constraints
- Delete rule

Tasks

- Creating and modifying foreign keys

Constraints

Use the **Constraints** box to create or modify foreign keys, which determine the relationship between tables.

Fields

⇑ Constraint list

The list box shows the existing constraints for the table.

To create a constraint, click &lt.NEW>, define the constraint, and click **Create** .

To modify a constraint, select a constraint from the list, change its definition, and click **Modify** .

To delete a constraint, select a constraint from the list and click **Delete** .

⇑ Constraint name

The **Constraint name** entry field shows the name of the currently selected constraint.

⇑ Parent table

The **Parent table** entry field shows the table referenced by the constraint.

When you want to create a new constraint, select **Search** to search for the appropriate parent table among the list of existing tables.

⇑ Parent table columns

The columns of the referenced parent table's primary key are listed here.

⇑ Matching columns

Here you see the matching columns.

Parent table columns and foreign key columns of the current, dependent table must match in the following aspects:
-Number of columns
-Data types
-Field procedures

Delete Rule

Certain constraints hold for the foreign keys in the dependent table when the DELETE operation is to be applied to the referenced parent table. The dependent table is the table under design.

Restrict	No rows will be deleted in the dependent table.
Cascade	The corresponding rows will also be deleted in the dependent table.
Nullify	The corresponding rows in the dependent table will be set to null (if they are nullable).
No action	An error occurs, and no rows are deleted in the dependent table. (DB2/390 Version 5 or higher.)

Checks Page

This page enables you to define and maintain the check constraints for one or more columns of a table. Supported for DB2/390 Version 4 or higher.

List

- List of check constraints

Fields

- Constraint name
- Check condition

Push Buttons

- Create
- Modify
- Delete

Indexes Page

On this page, you can optimize the access path by creating indexes on selected columns.

Fields

In the **Indexes** list box, all of the table's indexes are listed.

In the **Columns of selected index** list box, the columns belonging to the selected index are displayed.

Push Buttons

Select **Create** to create a new index.

Select **Open** to open the notebook of the selected index.

Select **Delete** to delete the selected index from the list of indexes defined on the table object.

Tasks

⇑ Creating indexes

Table Space Page

On this page, you can assign the table to an appropriate table space and database.

Fields

The currently assigned table space is listed in the **Table space** entry field, and the currently assigned database is listed in the **Database** entry field.

Push Buttons

Select **Open** to open the notebook of the listed table space or database.

When you want to assign a new table space or a new database, select **Search** to search for an appropriate table space or database among the list of existing ones.

Tasks

⇑ Assigning the table to a table space and a database

Partitions Page

Use this page to divide the table into partitions.

Table partitions are not a DB2 concept. They can be used to enable the DataAtlas Designer function of proposing partitioned table spaces and indexes. The concept of partitioning is especially useful for large tables.

A partition is made up of a concatenation of key ranges of selected columns.

A table's data load and work load will be entered per partition.

Fields

Select the appropriate **Number of partitions.** The number is listed in the corresponding field.

All of the columns available for the creation of partitions are listed in the **Available** partition columns list box.

The columns selected for the creation of partitions are listed in the **Selected** partitions columns list box.

The table of **Partitions** lists the created partitions. These partitions are identified by number. For each partition column, the target value delimiting its range (UPPER value) is listed.

Push Buttons

Select **Accept** to accept the specified number of partitions.

Select **Add** to add the selected columns to the list of partition columns.

Select **Remove** to remove the selected columns from the list of partition columns.

Select **Open** to specify range values for the columns of the partition.

A Table Partitions window displays. The number of the partition is listed. For each column, you specify the target UPPER value delimiting its range. The concatenation of specified column ranges determines the table partition.

Tasks

⇑ Creating table partitions
⇑ Modifying table partitions

Data Load Page

Together with the work load values, the data load values determine the adequate configuration of a table's access and storage structures.

Many of the design support functions DataAtlas Designer offers make use of the data load information. The most important value here is the value for the *initial number of rows.*

A read-only table shows the data load as specified for the table object. The data load is specified for each table partition. The partitions are identified by number. If no partitions are specified, the partition covers the entire table.

Fields

The following data load information is specified for each partition:

⇑ Confidence
⇑ Maintenance period (weeks)
⇑ Security option
⇑ Number of rows
⇑ Growth rate
⇑ Growth period
⇑ Delete rate
⇑ Delete period

Push Buttons

Select **Change...** to specify or change the data load information for the selected partition.

Confidence Factor

Valid values for the confidence factor are:

⇑ poor
⇑ medium
⇑ high
   This value specifies how confident you are that you have entered the proper estimation values.
The value has an effect on the secondary quantity (SECQTY) space calculations for table space and index space.

If the confidence factor is low, more secondary quantity space is allocated.

Maintenance Period

Based on the maintenance period and the growth and delete rate, the space allocations are calculated.

If you select a longer maintenance period, a higher amount of space will be proposed.

Valid values: 1 - 52 (unit = week).

Security Option

This option is only relevant for DataAtlas Designer design support. It affects the ERASE option proposal for table space and index.

If your data is security sensitive, set the security option.

In this case, DataAtlas Designer proposes to set the ERASE option for the table's table space and its indexes.

If your data is not security sensitive, do not set the security.

In this case, DataAtlas Designer proposes not to set the ERASE option.

Initial Number of Rows

This value specifies the estimated initial number of rows of the table.

Recommendation:

This value is required for most of the DataAtlas Designer's design support actions and should, therefore, always be specified.

Growth Rate

This value is an integer that specifies the estimated amount of rows by which the table will grow per growth period.

Growth Period

Valid values here are:

⇑ Day
⇑ Week
⇑ Month
⇑ Quarter

Delete Rate

This value is an integer that specifies the estimated amount of rows that will be deleted per delete period.

Delete Period

Valid values here are:

⇑ Day
⇑ Week
⇑ Month
⇑ Quarter

Work Load Page

Together with the data load values, the work load values determine the adequate configuration of a table's access and storage structures. DataAtlas Designer uses the workload information for index proposals and free space calculations.

A read only table shows the work load as specified for the table object. The work load is specified for each partition, which is identified by number. If no partitions are specified the partition covers the entire table.

Fields

The following work load information is specified for each partition:

⇑ Concurrency
⇑ Frequency for the following operations:
  -Select
  -Update
  -Delete
  -Insert
  -Unload/Reload
  -Reorg
  -Alter Table

Push Buttons

Select **Change...** to specify or change the work load information for the selected partition.

Concurrency

The concurrency value specifies the amount of concurrent access to the table. If partitions have been defined for the table, the concurrency value specifies the concurrent access to a table partition.

Valid values for concurrency are:

⇑ low
⇑ medium
⇑ high

Frequency

The frequency values are relative frequencies that specify how often an operation is used on the table.

Valid values for frequency range from 0 - 10, representing the scale low - medium - high.

If the frequency value is specified as 0 for the operations Update, Delete, and Insert, the table is treated as a read-only table.

DB2 Actuals Page

On this page, you can extract the most important statistic values of the table from the DB2 catalog. Together with the statistic values of the table, the statistic values of all the table's columns and assigned indexes are retrieved.

Fields

⇑ DB2 statistic values for the table (SYSIBM.SYSTABLES):

- Status
- Record Length
- Number of rows
- Percentage of rows
- Percentage of total pages
- Number of pages
- Last RUNSTATS time
- Last RUNSTATS date

⇑ Database alias

Push Buttons

Select **Search** to search for the database that contains the DB2 catalog from which you want to extract the statistic values.

Select **Extract statistics** to extract the current DB2 values of the table, and of the columns and the indexes that belong to the table.

Note:

No data is available for extraction as long as the object has not been implemented.

Status

A table can have one of the following states:

I	The table lacks a primary index.
X	The table has a primary index.
blank	The table has no primary key.

DB2 catalog name: SYSTABLES.STATUS.

Number of Rows

This value reflects the current number of rows of the table under design.

DB2 catalog name: SYSTABLES.CARD.

Percentage of Rows

This value reflects the percentage of active rows multiplied by 100, compressed within the total number of active rows in the table.

DB2 catalog name: SYSTABLES.PCTROWCOMP.

Number of Pages

This value reflects the total number of pages which hold rows of the table.

DB2 catalog name: SYSTABLES.NPAGES.

Percentage of Total Pages

This value reflects the percentage of all of the table space's pages which hold rows of the table.

DB2 catalog name: SYSTABLES.PCTPAGES.

Record Length

This value reflects the maximum length of any record in the table.

DB2 catalog name: SYSTABLES.RECLENGTH.

Last RUNSTATS time

This value reflects the time when RUNSTAT was last invoked to update the statistics.

DB2 catalog name:

SYSINDEXES.STATSTIME.

SYSINDEXPART.STATSTIME.

SYSTABLES.STATSTIME.

SYSTABLEPART.TIME.

Last RUNSTATS date

This value reflects the date when RUNSTAT was last invoked to update the statistics.

DB2 catalog name:

SYSINDEXES.STATSTIME.

SYSINDEXPART.STATSTIME.

SYSTABLES.STATSTIME.

SYSTABLEPART.TIME.

Database Alias

This field shows the name of the database that contains the DB2 catalog from which you want to extract the statistic values.

Designs Page

On this page, you can:

⇑ Select design functions to assess the table

⇑ Assign the table to a physical design

Fields

In the **Physical designs** list box, you see the physical designs to which the table object currently belongs.

Push Buttons

Select **Inform** to get information about potential design problems in the table.

Select **Validate** to check the correctness of the table's design.

Select **Propose** to get suggested design improvements that you can execute.

Select **Open** to open the selected physical design.

Select **Search** to search for an appropriate physical design among the list of existing physical designs, when you want to assign the table to a physical design.

Select **Remove** to remove the table object from the selected physical designs. The selected physical designs are deleted from the **Physical designs** list.

Physical Designs

This list box shows the physical designs to which the object you are currently designing belongs.

The physical design covers all the database objects necessary for an efficient operation of a DB2/390 database. Efficiency is reached by optimum storage and access structures for the tables of a design. For a DB2/390 database, the following database objects are contained in a physical design:

⇑ Table
⇑ Index
⇑ Table space
⇑ Database
⇑ Storage group

You can open a physical design to look at all of the objects belonging to it.

When you want to assign the present design object to a physical design you can search for an existing physical design to use.

Also, you can remove the present design object from selected physical designs.

General Page

Use this page to specify general information about a DB2/390 table object.

Once you have stored the object in the TeamConnection database, you cannot change the relational database qualifiers. The other specifications can be changed at any time.

Fields

⇑ Relational Database Qualifiers:
- System
- Creator

Select **Search** to search for a list of items that can be entered into the corresponding field. If you select an item from the list, it will appear in the field.

⇑ Name
⇑ Variation
⇑ Revision
⇑ Component

Select **Search** to see a list of components represented in the TeamConnection database. When you select a component name from the list, it is put in the Component field.

⇑ Description
⇑ History

Name

This field contains the name of the DB2/390 table object.

The value for the name must be a valid DB2 identifier .

Required?: Yes.

DB2/390 Table Space Notebook

Use the table space notebook to maintain and design the selected table space object. The notebook pages indicate the different design areas.

You can get design support for a table space from anywhere in the notebook.

Pages

- Options
- Partitions
- Storage
- Tables
- Buffer Pool
- Design Info
- DB2 Actuals
- Designs
- General

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

Menu-Bar Choices

- Table space (Object management)
- Designer
- Window
- Help

Tasks

- Designing a table space object

Designer (for Table Space)

This menu-bar choice displays a pull-down menu with the types of design support for table spaces :

⇑ **Inform - Check completeness.** Get information about potential design problems in the table space.

⇑ **Validate - Check correctness.** Check the correctness of the table space's design.

⇑ **Propose - Suggest improvements.** Get suggested design improvements that you can execute.

Options Page

Use this page to specify the type and other options for the selected table space object.

Fields

⇑ Type
  - Simple
  - Segmented
    - Segment size
  - Partitioned
    - Number of partitions
  - LARGE
  - LOCKPART

⇑ MAXROWS
⇑ Password
⇑ Close option
⇑ LOCKSIZE
⇑ LOCKMAX
⇑ CCSID

Type

There are three types of table spaces:

- **Simple.**

  The table space is composed of pages. Each page can contain rows from many tables.

  Note: DataAtlas Designer does not propose simple table spaces, because in most cases segmented or partitioned table spaces are to be preferred. However, you can always assign a simple table space directly.

- **Segmented.**

  The table space is composed of groups of pages called segments. Each segment holds rows of a single table.

  The **Segment size** indicates how many pages are to be assigned to each segment.

- **Partitioned.**

  The table space can only contain a single table. The table space is subdivided into partitions based on the key range of a nominated partitioning index which is required.

  A partitioned table space is an advantage for large tables. Each partition can be maintained separately (utilities COPY, RECOVER, REORG).

  The **Number of partitions** cannot exceed 64 partitions.

- **LARGE.**

  A **large** table space is a partitioned table space that can hold more than 64 gigabytes of data, either compressed or uncompressed. It can have up to 254 partitions, with a maximum partition size of 4 gigabytes.

  If you check **LARGE** , you have to specify a number of partitions. If you don't check it and there are more than 64 partitions, the table space will nevertheless have the attributes of a large table space.

  This check box is supported for DB2/390 Version 5 or higher.

- **LOCKPART.**

  Use this check box to indicate whether you want selective partition locking (SPL) of a partitioned table space. If you check the box and the conditions needed for SPL are met, only partitions that are accessed are locked. If you don't check the box or SPL conditions aren't met, every partition of the table space is locked. You cannot check this box if the **LOCKSIZE** field specifies TABLESPACE.

  This check box is supported for DB2/390 Version 5 or higher.

MAXROWS

This field specifies the maximum number or rows that DB2 can place on a data page. You cannot specify a MAXROWS value for a table space in a work file database.

This field is supported for DB2/390 Version 5 or higher.

Required?: No.

Limits: 1 to 255.

Default: 255.

Password

This field specifies a master level password sent to access method services when the data sets of the table space are used by DB2.

The value for the password is a valid DB2 short identifier.

If you use a storage group, this password protects the data sets as well as it is the one passed to access method services when the data sets are used by DB2. If you are defining the data sets yourself, the password is only the one that is passed on to VSAM (virtual storage access method); you need to define separately the password that protects the data sets.

All password protected data sets of the table space must have the same password.

LOCKSIZE

The locking size for table space can have the following values:

ANY	This allows DB2 to choose a lock of the smallest convenient size (usually a page lock). This value is the default.
PAGE	With this option, an application process acquires page locks, plus table locks (in a segmented table space) and table space locks of modes that permit page locks (IS, IX, or SIX).
ROW	With this option, all the indexes defined on tables in this table space must be Type 2 indexes. This option is supported for DB2/390 Version 4 or higher.
TABLE	With this option, table locks are acquired for tables contained in segmented table spaces. Table locks are either S, U, or X.
TABLESPACE	With this option, no page locks are acquired within the table space. If the table space is segmented, no table locks are acquired. Table locks are either S, U, or X.

Note:

S	SHARE
U	UPDATE
X	EXCLUSIVE
IS	INTENT SHARE
IX	INTENT EXCLUSIVE
SIX	SHARE with INTENT EXCLUSIVE

LOCKMAX

Check this box to enable the **System** and **Integer** buttons. They let you specify the maximum number of page or row locks a program can hold simultaneously in the table space.

If you select the **System** button, the maximum number of locks is taken from the value of LOCKS PER TABLE(SPACE) on the DSNTIPJ installation panel. If you select the **Integer** button, specify the maximum number in the adjacent field. The largest number you can specify is 2147483647, but if you specify 0, an unlimited number of locks are allowed.

This field is supported for DB2/390 Version 4 or higher.

Partitions Page

Use this page to specify the type and storage information for table space partitions.

A read-only table shows the existing partitions and their specifications.

Fields

The following information is specified for each partition:

⇑ Compress

Specifies whether data compression applies to the rows of the table space or table space partition.

⇑ Partition

Shows the number by which partitions are identified.

⇑ Type

Indicates whether the data sets for the table space are to be managed by the user (VCAT) or managed by DB2 (STG - Storage group).

⇑ VCAT

Specifies the volume catalog name.

⇑ Storage group

Specifies the storage group name.

⇑ PRIQTY

Specifies the minimum primary space allocation for a DB2-managed data set.

⇑ SECQTY

Specifies the minimum secondary space allocation for a DB2-managed data set.

⇑ ERASE

Specifies whether DB2-managed data sets are to be erased when they are deleted during the execution of a utility or when the index is dropped.

⇑ FREEPAGE

Specifies how often a free page must be reserved in the table space.

The default is 0, leaving no free pages.

⇑ PCTFREE

Specifies the percentage of space that must be left empty in each page.

⇑ GBPCACHE

Specifies what pages of the table space or partition are written to the group buffer pool in a data sharing environment. (DB2/390 Version 4 or higher.)

Push Buttons

Select **Create** to create new table space partitions.

A window for creating new table space partitions displays where you can specify type and storage information for the new table space partition.

Select **Open** to open a selected table space partition for viewing or changing the partition's specifications of type and storage information.

Select **Delete** to delete the selected partition.

Note: DB2 requires that the number of table space partitions match the number of index partitions.

If you have used table partitions for the creation of table space partitions, and if you want to make any changes, change the table partitions.

FREEPAGE

This integer value specifies how often to leave a page of free space when the table space or table space partition is loaded or reorganized. You must specify an integer in the range 0 to 255. If you specify 0, no pages are left as free space. Otherwise, one free page is left after every n pages, where n is the specified integer.

The default is 0, leaving no free pages.

Do not use this option for a table space in DSNDB07.

Note: For this value you can request a proposal from DataAtlas Designer.

PCTFREE

This integer specifies what percentage of each page to leave as free space when the table space or table space partition is loaded or reorganized. The integer can range from 0 to 99.

The default is 5.

Do not use this option for a table space in DSNDB07.

Note: For this value you can request a proposal from DataAtlas Designer.

Compress

This option specifies whether data compression applies to the rows of the table space or table space partition.

Storage Page

Use this page to specify the necessary storage information for the table space. This includes the specification of free space as well as the specification of whether the data sets for the table space are to be managed by the user (VCAT option) or managed by DB2 (Storage group option). You make these specifications by using the items within the **Using block** (for DB2/390 Version 4 or higher), **Free block** , and **GBPCACHE block** (for data sharing environments, DB2/390 Version 4 or higher). Checking the box beside a block enables the items within it.

If you omit the **Using block** information, the data sets are managed by DB2 on volumes listed in the default storage group of the table's database; primary and secondary quantity (PRIQTY, SECQTY) and the **Erase** check box assume their default values.

Check Box

⇑ **Compress** : Specifies whether data compression applies to the rows of the table space. If the option is set, the rows are not compressed until the LOAD or REORG utility is run on the table in the table space.

Blocks

⇑ Using block
- **VCAT**
  - **Catalog name**

- **Storage group**
  - Name

    Select **Search** to search for a list of existing storage groups to select from.
  - **PRIQTY**
  - **SECQTY**

    Select **Convert** to convert primary and secondary quantity values from kilobytes to the number of cylinders or tracks.
  - **Erase**

⇑ Free block
- **FREEPAGE**
- **PCTFREE**

⇑ GBPCACHE
- **Changed**
- **All**

Tables Page

On this page, you can assign tables to the table space.

Fields

The currently assigned tables are listed in the list box.

Push Button

Select **Open** to open the notebook of a selected table.

Buffer Pool Page

Use this page to identify the buffer pool to be used for the table space and to determine the page size of the table space. For 4KB page buffer pools, the page size is 4KB. Otherwise, the page size is 32 KB.

Fields

The buffer pool which is currently used by the table space is displayed. Further buffer pools are contained in the list box. Select one of them if you want to use another buffer pool.

Note: For selection of the appropriate buffer pool, you can request a proposal from DataAtlas Designer.

Design Info Page

Use this page to specify the necessary design information of a table space.

Fields

- Keep in main storage
- Usage intent

Keep in Main Storage

This table space option specifies whether tables assigned to the table space are to be kept in main storage.

Option set:	Keep in main storage
Option not set:	Do not keep in main storage

It is advisable to keep table space in main storage when it contains one or more small tables which are accessed very often. Instead of checking them in and out each time they are accessed, its data is better kept in main storage in a fixed buffer pool.

If you want to keep in main storage a table space that contains one or more small tables, make sure that the buffer pool you select for the table space is reserved exclusively for this table space, and is sufficiently large to hold this table space.

Usage Intent of Table Space

The usage intent of a table space can be one of the following:

QMF	Table space is used for a QMF (Query Management Facility) application.
Batch	Table space is used for a batch application.
Test	Table space is used for test purposes, and not for production.
Random	Table rows are distributed randomly in the table space.

Note: Among others, DataAtlas Designer uses the information on a table space's usage intent to calculate the values for PCTFREE (table space and index) and for SUBPAGES (index).

DB2 Actuals Page

On this page, you see the values of the table space as currently implemented in the DB2 database.

Fields

⇑ DB2 statistic values for the table space (SYSIBM.SYSTABLESPACE):
- Status
- Number of active pages
- Space (KB)
- Implicit creation

⇑ DB2 statistic values for the table space partitions (SYSIBM.SYSTABLEPART):
- Number of rows
- Number of far rows
- Number of near rows
- Active tables
- Dropped tables
- Space
- Saved pages
- RUNSTATS time
- RUNSTATS date

⇑ Database alias

Push Button

Select **Search** to search for the database that contains the DB2 catalog from which you want to extract the statistic values.

Select **Extract statistics** to extract the current DB2 values from the catalog.

Status

The table space can have one of the following states:

A	Available.
C	Definition is incomplete because no partitioned index has been created.
P	Table space is in a check pending state.
S	Table space is in a check pending state with the scope less than the entire table space.
T	Definition is incomplete because no table has been created.

DB2 catalog name: SYSTABLESPACE.STATUS.

Number of Active Pages

This integer value denotes the number of active pages in the table space.

DB2 catalog name: SYSTABLESPACE.NACTIVE.

Implicit Creation

This value indicates whether a table space is implicitly created.

Y	Implicit creation.
N	No implicit creation.

DB2 catalog name: SYSTABLESPACE

Number of Rows

This integer value denotes the number of rows in the table space or table space partition.

DB2 catalog name:

SYSTABLESPACE.CARD.

SYSTABLEPART.CARD.

Number of Far Rows

This integer value denotes the number of rows that have been relocated far from their original page.

DB2 catalog name: SYSTABLEPART.FARINDREF.

Number of Near Rows

This integer value denotes the number of rows that have been relocated close to their original page.

DB2 catalog name: SYSTABLEPART.NEARINDREF.

Active tables

This integer value denotes the percentage of space occupied by rows with data from active tables.

DB2 catalog name: SYSTABLEPART.PERCACTIVE.

Dropped Tables

This integer value denotes the percentage of space occupied by rows from dropped tables.

The value is 0 for segmented table spaces.

DB2 catalog name: SYSTABLEPART.PERCDROP.

Saved Pages

This integer value denotes the percentage of pages multiplied by 100, which are saved in the table space or table space partition as a result of using data compression.

DB2 catalog name: SYSTABLEPART.PAGESAVE.

Designs Page

On this page, you can assign the table space to a physical design.

Fields

In the **Physical designs** list box, you see the physical designs to which the table space object currently belongs.

Push Buttons

Select **Inform** to get information about potential design problems in the table space.

Select **Validate** to check the correctness of the table space's design.

Select **Propose** to get suggested design improvements that you can execute.

Select **Open** to open the selected physical design.

Select **Search** to search for an appropriate physical design among the list of existing physical designs, when you want to assign the table space to a physical design.

Select **Remove** to remove the table space object from the selected physical designs. The selected physical designs are deleted from the **Physical designs** list.

General Page

Use this page to specify general information about a DB2/390 table space object.

Once you have stored the object in the TeamConnection database, you cannot change the relational database qualifiers. The other specifications can be changed at any time.

Fields

⇑ Relational Database Qualifiers:
- System
- Database

Select **Search** to search for a database to associate with this table space. When you make a selection, the **System** field is filled in along with the **Database** field.

⇑ Name
⇑ Variation
⇑ Revision
⇑ Component

Select **Search** to see a list of components represented in the TeamConnection database. When you select a component name from the list, it is put in the Component field.

⇑ Description
⇑ History

System

This field contains the name of the location of the DB2 system.

Required?: Yes, for all DB2 objects.

Creator

This field contains the name of the creator of the DB2 object. It is filled in by using the **Search** push button.

Required?: Yes, for DB2/390 tables, indexes, and views, and for DB2 aliases or synonyms.

Database

This field contains the name of the DB2 relational database. The value of the name must be a valid DB2 identifier .

If no database name is entered, the default database name is used. The DB2 default is: DSNDB04.

Required?: Yes, for DB2/390 table spaces.

Name

This field contains the name of the DB2/390 table space object.

The value for the name must be a valid DB2 long identifier.

Required?: Yes.

Variation

This field contains a qualifier that becomes part of the full name of the object in the TeamConnection database. The qualifier can be useful in distinguishing between similar objects that have different uses.

Required?: No.

Default: Initialized from the profile.

Revision

This field contains a qualifier that becomes part of the full name of the object in the TeamConnection database. The qualifier can be useful in distinguishing between similar objects that have to be available at the same time.

Required?: No.

Default: Initialized from the profile.

Component

This field contains the name of the component to which the DB2 object belongs.

A component is a logical group of development data within a TeamConnection family.   Identifying a component is a means of controlling access to development data.

Required?: Yes.

Default: Initialized from the profile.

Description

This field contains a text description of the DB2 object.

Required?: No.

History

This read-only field shows, by date and time, when the object was last modified.

DB2/390 Table Space Partitions Window

Use this window to view or specify storage information for a table space partition. This includes specifying free space and whether the data sets for the partition are to be managed by the user (VCAT option) or managed by DB2 (Storage Group option). You make these specifications by using the items within the **Using block** (for DB2/390 Version 4 or higher), **Free block** , and **GBPCACHE block** (for data sharing environments, DB2/390 Version 4 or higher). Checking the box beside a block enables the items within it.

If you omit the **Using block** information, the data sets are managed by DB2 on volumes listed in the default storage group of the table's database; primary and secondary quantity (PRIQTY, SECQTY) and the **Erase** check box assume their default values.

Check Box

⇑ **Compress** : Specifies whether data compression applies to the rows of the table space partition. If the option is set, the rows are not compressed until the LOAD or REORG utility is run on the table in the table space partition.

Blocks

⇑ Using block
- **VCAT**
  - **Catalog name**
- **Storage group**
  - Name

    Select **Search** to search for a list of existing storage groups to select from.
  - **PRIQTY**
  - **SECQTY**

    Select **Convert** to convert primary and secondary quantity values from kilobytes to the number of cylinders or tracks.
  - **Erase**

⇑ Free block
- **FREEPAGE**
- **PCTFREE**

⇑ GBPCACHE
- **Changed**
- **All**

DB2/390 View Notebook

Use this notebook to create and maintain a DB2/390 view object.

Pages

- Definition
- Designs
- General

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

Menu-Bar Choices

- View (Object management)
- Window
- Help

Definition Page

Use this page to define a view.

You formulate the SQL SELECT statement that specifies which columns are to be accessed from which tables, you specify whether the view is to be used with or without the Check Option, and you can list the columns contained in the view.

Fields

⇑ Subselect
⇑ Check option
⇑ Comment

This field gives you room for your own notes and comments on the view object.

⇑ Label
⇑ Columns (optional)

Subselect

Use this entry field to formulate the SQL SELECT statement that specifies which columns are to be accessed from which tables for the view you want to create.

Columns of a View

Here, you can list the columns that participate in the view. Filling the list is optional.

However, DB2 requires that you specify the column names if the result table of the subselect statement has duplicate column names or an unnamed column.

Fields

⇑ Name
⇑ Comment

This field gives you room for your own notes and comments on the view column.

⇑ List of view columns

Select **Add** to add the column you entered into the Name field to the list of view columns.

Select **Modify** to put a changed view column back to the list of view columns.

Select **Remove** to delete the selected columns from the list of view columns.

Name of View Column

This field contains the name of the view column.

The value for the name must be a valid DB2 short identifier .

In the list of view columns the name of the view column is shown in its complete internal representation.

List of View Columns

This list contains all the columns of the view.

You may fill in this list of view columns to match the view columns specified in the SQL SELECT statement.

The list of columns names must consist of as many names as there are columns in the result table of the subselect statement.

The list is not created automatically.

Designs Page

On this page, you can assign the view to a physical design.

Fields

In the **Physical designs** list box, you see the physical designs to which the view object currently belongs.

Push Buttons

Select **Open** to open the selected physical design.

Select **Search** to search for an appropriate physical design among the list of existing physical designs, when you want to assign the view to a physical design.

Select **Remove** to remove the view object from the selected physical designs. The selected physical designs are deleted from the **Physical designs** list.

General Page

Use this page to specify general information about a DB2/390 view object.

Once you have stored the object in the TeamConnection database, you cannot change the relational database qualifiers. The other specifications can be changed at any time.

Fields

⇑ Relational Database Qualifiers:
- System
- Creator

Select **Search** to search for a list of items that can be entered into the corresponding field. If you select an object from the list, it will appear in the field.

⇑ Name
⇑ Variation
⇑ Revision
⇑ Component

Select **Search** to see a list of components represented in the TeamConnection database. When you select a component name from the list, it is put in the Component field.

⇑ Description
⇑ History

Name

This field contains the name of the DB2/390 view object.

The value for the name must be a valid DB2 identifier .

Required?: Yes.

DB2 UDB Column Notebook

This notebook enables you to define the data attributes of a new column or modify the attributes of an existing column.

Pages

- Definition
- Option

Push Buttons

- OK
- Save
- Cancel
- Help

Definition Page

This page enables you to define and update basic information for a column, like the type of data it contains and the data's length, precision, and scale. It also lets you connect a shareable data element to the column.

Fields

- Name
- Comment
- Shareable Data Element Name
- Length

Lists

- Type
- Precision
- Scale

Push Buttons

- Search
- Open
- Remove

Check Boxes

- Bit data
- LOB options

Name

This field identifies the name of a column.

Required?: Yes.

Limit: 18 characters.

The column name must be in a valid format. Refer to the DB2 UDB reference manual for information on name restrictions.

Comment

This field contains a comment for the column.

Limit: 254 characters.

Shareable Data Element Name

This read-only field contains the name of the shareable data element you have chosen to define the column.

Length

This field contains the maximum length of the column's data. The adjacent drop-down box is enabled if the data type is BLOB, CLOB, or DBCLOB. In this case you can select K (kilobytes), M (megabytes), G (gigabytes), or the blank field from the box. K, M, and G designate the units that apply to the number in the **Length** field. The blank field designates that the length is in byte units.

Default: 1.

Type

This drop-down list contains all the valid data types for a column. After one data type is chosen for the column, only the appropriate fields will be enabled for you to update. For example, if the **CHAR** data type is chosen, only the Length field will be enabled.

Required?: Yes.

This field will be disabled if the column is being used in the primary key or in a foreign key.

Precision

This read-only field contains the precision value of a **DECIMAL** number.

Default: 5.

Limit: 31.

Scale

This read-only field contains the scale value of a **DECIMAL** number.

Default: 0.

Limit: The precision of the decimal number or 31.

Search

Click this button to search for a shareable data element for this column. The search will be qualified by values in the fields that define the column.

Open

Click this button to open the Shareable Data Element notebook named in the adjacent field.

Remove

Click this button to remove a shareable data element from the column definition. Values in the fields that define the column will be cleared.

Bit Data

Check this box if you want the contents of the column to be treated as bit (binary) data for exchange with other systems.

This box is enabled only for **CHAR, VARCHAR**, and **LONG VARCHAR** columns.

LOB Options

The **LOGGED** and **COMPACT** check boxes are enabled only for a LOB data type. Check the **LOGGED** box if you want changes to the column to be logged. Check the **COMPACT** box if you want the LOB column to use minimal disk space. There may be a performance trade-off if you check the box and later append to the column.

Option Page

This page enables you to define the column's NULL and DEFAULT attributes. You can also enter an extended description of the column here.

Check Boxes

Null Not Allowed
DEFAULT

Field

Description

Null Not Allowed

Check this box to prevent the column from containing null values.

Default: Unchecked.

DEFAULT

Check this box to specify a default value for the column if no value has been inserted. Select one of the options below this box to specify the default value.

Default: Unchecked.

The meanings of the options are:

By data type	The default depends on the data type of the column.
Constant	The radio button enables the adjacent field, which specifies a default value for the column. The value must conform to the rules for assigning a value to a column.
Datetime	The default value will be the value of the datetime special register. The adjacent field will describe the value as CURRENT DATE, CURRENT TIME, or CURRENT TIMESTAMP, depending on the column's data type.
USER	The default is the value of the USER special register.
NULL	The default is NULL. This choice is not available if the **NOT NULL** box is checked on this page.

Description

This field contains a text description of the column.

Required?: No.

Limit: 4096 characters.

DB2 UDB Database Notebook

This notebook enables you to name and describe a DB2 UDB database object and save it in the TeamConnection database.

Menu-Bar Choices

- Database
- Window
- Help

Page

- General

Push Buttons

- OK
- Save
- Cancel
- Help

Database

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

General Page

This page enables you to name the DB2 UDB database and write a description of it. The **System** field cannot be modified for an existing database.

Fields

- System
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the DB2 UDB database.

Required?: Yes.

Variation

This field contains the name of a variation.

A variation name becomes part of the full name of an object in the TeamConnection database.   It can be useful in distinguishing between similar objects that have different uses.

If an object is related to a higher-level object, it takes the variation name, if any, of the higher-level object.   So, for example, a relational database object will automatically be assigned the variation name given to the higher-level system object when it was defined.

Required?: No.

Revision

This field contains the name of a revision.

A revision is a qualifier that becomes part of the full name of an object in the TeamConnection database. This qualifier can be useful in distinguishing between similar objects that have to be available at the same time.

If an object is related to a higher-level object, it takes the revision name, if any, of the higher-level object. So, for example, a relational database object will automatically be assigned the revision name given to the higher-level system object when it was defined.

Required?: No.

DB2 UDB Index Notebook

This notebook enables you to create or modify an index for a DB2 UDB table.

Pages

- Definition
- Options
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Definition Page

This pages enables you to:

⇑ Search the TeamConnection database for a table whose index you want to create or modify

⇑ Specify which columns you want to be part of the index

⇑ Prevent index columns from having duplicate values

⇑ Specify the sorting order of index columns

Table

- Table Name
- Search
- Open

Lists

- Available columns
- Index columns

Push Buttons

- Add
- Remove

Radio Buttons

- Unique Constraint
- Column Order

Table Name

This read-only field identifies the table this index is based on. You cannot change this field for an existing index. For a new index, you choose a table by selecting the Search push button and then selecting a table from the TeamConnection database.

Required?: Yes.

Available Columns

This list is initialized after you select a table for the index. All the column names in the table are listed.

To add a column to the index, select a column in this list and click Add .

Index Columns

This list contains the columns that defines this index. To add a column to this list, select a column from the Available columns list and click Add. To remove a column from this list, select a column in this list and click Remove. There must be at least one column in the index.

Required?: Yes.

Search

Click this button to see a list of tables that belong to the relational system this index belongs to. (The name of the relational system is in the **System** field on the **General** page.)

Open

Click this button to open the properties notebook for the table of this index.

Add

Click this button to add a column to the index. The button will remain disabled until a column is selected from the Available columns list.

Remove

Click this button to remove a column from the index.

Unique Constraint

Click **Yes** if you want to prevent the table from containing rows with the same value of the index key. Click **No** if you don't want to prevent it.

Default: No.

Column Order

Select the **Ascending** button to sort the index columns in ascending order. Select the **Descending** button to sort them in descending order.

Default: Ascending.

Options Page

On this page you can connect the index to or disconnect it from an existing physical design.

List

- Physical designs

Push Buttons

- Search
- Open
- Remove

Physical Designs

This is a list of physical designs to which this index belongs. You can connect the index to a new physical design by selecting the adjacent **Search** push button and a physical design from the resulting list. The name of the selected physical design will be added to the list on this page.

Required?: No.

Search

Click this button to find a physical design to which you can connect this index.

Open

Click this button to open a **Physical Design** notebook for a selected physical design.

Remove

Click this button to remove the selected physical design from the list and disconnect this index from it.

General Page

This page enables you to enter general information about a DB2 UDB index. The **System** , **Schema** , and **Database** fields cannot be modified for an existing index.

Fields

- System
- Schema
- Database
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Schema

This field contains the user ID under which the index was created. When a schema is selected by using the **Search** push button, both this field and the **System** field are initialized.

Required?: Yes.

Name

This field contains the access name of the DB2 UDB index.

Required?: Yes.

DB2 UDB Physical Design Notebook

This notebook enables you to open the properties notebooks of the objects that belong to this DB2 UDB physical design.

The notebook's pages correspond to the objects that can be in a DB2 UDB physical design - tables, table spaces, indexes, and views.

Menu-Bar Choices

- Physical design
- Window
- Help

Pages

- General
- Index
- Table
- Table Space
- View

Push Buttons

- OK
- Save
- Cancel
- Help

General Page

Use this page to specify general information about a DB2 UDB physical design.

Fields

- Relational design
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Index Page

This page enables you to view the list of indexes that belong to the physical design and open the properties notebook for an entry selected from the list.

List

- Indexes

Push Button

- Open
- Search
- Remove

Indexes

This list identifies the indexes that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook.

Table Page

This page enables you to view the list of tables that belong to the physical design and open the properties notebook for any entry selected from the list.

List

- Tables

Push Button

- Open
- Search
- Remove

Tables

This list identifies the tables that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook.

Table Space Page

This page enables you to view the list of table spaces that belong to the physical design and open the properties notebook for any entry selected from the list.

List

- Table spaces

Push Button

- Open
- Search
- Remove

Table spaces

This list identifies the table spaces that belong to this physical design.

If you select an entry in the list, you can use the **Open** push button to open its properties notebook.

View Page

This page displays the list of views that belong to the physical design. You can open the properties notebook for an entry selected from the list.

List

- Views

Push Button

- Open
- Search
- Remove

DB2 UDB Table Notebook

This notebook enables you to define and maintain a DB2 UDB table in the TeamConnection database.

Pages

- Definition
- Primary Key
- Foreign Keys
- Checks
- Options
- Indexes
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Definition Page

This page enables you to specify or modify information about a table. It displays the name of the shareable table definition that it uses and allows you to enter a comment for it.   It also displays all the columns in the table. By using push buttons, you can create new columns or modify existing ones.

Fields

- Name (of Shareable Table Definition
- Comment

Table

- Columns

Push Buttons

For Shareable Table Definition:

- Open

For Columns table:

- Create
- Open
- Delete
- Move up
- Move down

Name (of Shareable Table Definition)

This field contains the name of the shareable table definition that this table uses. For a new table, DataAtlas will use the table name as a default for the shareable table definition name. If the name is not unique, DataAtlas will create a new name for the shareable table definition (usually a number will be concatenated to the end of the table name until a unique name is found).

Read-only?: Yes.

Note: On the Definition page of the **Shareable Table Definition** notebook, you can assign this table definition to a relational design. You have to take this action to use DataAtlas Modeler with the table definition.

Comment

This field contains a comment that describes this shareable table definition.

Limit: 256 characters.

Columns

The column container has all the columns in this table. To create a new column, click Create . To modify an existing column, select the column, then click Modify... , Delete , Move Up or Move Down .

Search

Click this button to search for existing shareable table definitions.

Open (Shareable Table Definition)

Click this button to open the notebook of the shareable table definition identified in the Name field. For a new table, this push button is disabled.

Create

Click this button to create a new column in the table.   This will open a **Column** notebook, where you can define the new column.

Open (Columns Table)

Click this push button to modify an existing column after you have clicked on the column in the **Columns** table. A **Column** notebook opens, enabling you to modify the column's attributes.

Note: If the column is part of the primary key or is used in a foreign key, you will not be able to modify the data attributes of that column. Also, if the column is part of the primary key, its null attribute cannot be changed to YES.

Delete

Click this button to delete an existing column after clicking on the column in the **Columns** table. You will be prompted to confirm the deletion.

Note: If the column is part of the primary key or a foreign key, or is used in an index, you cannot delete the column.

Move Up

Click this button to rearrange the order of the columns in the table after clicking on a column in the **Columns** table. The column will move up one position in the list or to the left one position in the table.

Note: If you are maintaining a table that exists in your database, reordering columns will have the effect of dropping the existing table and recreating a new one; the ALTER TABLE statement doesn't support column reordering.

Move Down

Click this button to rearrange the order of the columns in the table after you have clicked on a column in the **Columns** table. The column will move down one position in the list or to the right one position in the table.

Note: If you are maintaining a table that exists in your database, reordering columns will have the effect of dropping the existing table and recreating a new one; the ALTER TABLE statement doesn't support column reordering.

Primary Key Page

On this page you can give the primary key a name and choose which columns define the primary key for the table.

Fields

- Constraint name
- Comment

Lists

- Available columns
- Primary key columns

Push Buttons

- Add
- Remove

Constraint Name

This field can contain the name of the primary key.

Required?: No.

Limit: 18 characters.

Comment

This field can contain a comment about the primary key.

Required?: No.

Limit: 254 characters.

Available Columns

This list contains all the columns that qualify to be used in a primary key. The columns in the list are columns that are not used in the primary key already. They must be defined as NOT NULL or NOT NULL WITH DEFAULT and cannot have a data type of LOB, LONG VARCHAR, or LONG VARGRAPHIC.

Primary Key Columns

This list contains the columns that define the table's primary key.

You can add columns to the primary key by clicking on a column in the Available columns list and then selecting the Add push button.

You can remove columns from the primary key by clicking on a column in the **Primary Key columns** list and then selecting the Remove push button.

Limit: The number of identified columns must not exceed 16, and the sum of their lengths must not exceed 255.

Note: If the current table is referenced by other tables, their foreign keys are closely tied with this primary key. Modifying the primary key will have the effect of dropping all the foreign keys of the child tables. You will be prompted to continue in this case.

Add

Click this button to add a selected column from the Available columns list to the Primary Key columns list.

Remove

Click this button to remove a selected column from the Primary Key columns list and adds it to the Available columns list.

Foreign Keys Page

This page enables you to add and remove a foreign key. To add a foreign key, click &lt.NEW> in the foreign key list, define the new foreign key, and click **Create** . To remove a foreign key, select an existing foreign key and click **Delete** .

Lists

- List of foreign keys
- Primary key
- Matching columns

Fields

- Constraint name
- Comment
- Referenced table

Radio Buttons

- Delete rule
- Update rule

Push Buttons

- Create
- Delete
- Search

List of Foreign Keys

This list identifies all the foreign keys for this table. Select the entry &lt.NEW> if you want to create a new foreign key. The fields below the list will be cleared, enabling you to define it. Click **Create** when you are finished defining it.

Primary Key

This list contains the referenced table's primary key columns. For each primary key column of the referenced table, a corresponding matching column is displayed.

For a new foreign key, the list is initialized when a referenced table is selected. All the corresponding matching columns are blank.

Matching Columns

This drop-down list contains the columns used to define a foreign key.   For each primary key column of the referenced table, a corresponding matching column is shown.   For maintenance purposes, the drop-down list for each matching column contains columns qualified to be used as alternates.

For a new foreign key, the field above the drop-down list is blank. Opening the drop-down list displays all the columns in the table that can be used as a matching column for the corresponding primary key column.

Constraint Name

This field contains the name of a foreign key.

Required?: Yes.

Limit: 8 characters.

Comment

This field can contain a comment about the foreign key.

Required?: No.

Limit: 254 characters.

Referenced Table

This read-only field contains the name of the referenced table of the selected foreign key. For a new foreign key, you need to fill in the name of the referenced table by searching for it in the TeamConnection database.

Required?: Yes.

Note: For an existing foreign key, this field cannot be changed.

Delete Rule

Select the radio button representing the appropriate delete rule for the foreign key.

Default: No action.

Update Rule

Select the radio button representing the appropriate update rule for the foreign key.

Default: No action.

Create

Click this button to create a new foreign key after you have:

1. Clicked the entry marked &lt.NEW> in the list of foreign keys.

2. Entered a constraint name for the new foreign key.

3. Clicked the **Search** button next to the **Referenced table** field to search for a reference table.

4. Selected a table from the search result list.

5. Selected the delete rule and update rule you want.

Delete

Select an entry in the **List of foreign keys** and click this button to delete the selected entry.

Search

Click this button to see a list of tables that belong to the relational system this table belongs to. (The name of the relational system is in the **System** field on the **General** page.) When you select a table name from the list, it is put in the **Referenced table** field.

Checks Page

This page enables you to define and maintain the check constraints for one or more columns of a table.

List

List of check constraints

Fields

Constraint name
Comment
Check condition

Push Buttons

Create
Modify
Delete

List of Check Constraints

This list identifies all the check constraints for this table. Select the entry &lt.NEW> if you want to create a check constraint. The fields below the list will be cleared, enabling you to define it. Click **Create** when you are finished defining it.

Select the name of a check constraint if you want to update its definition. Then click **Modify** to apply your update.

Constraint Name

This field contains the name of the check constraint.

Required?: Yes.

Default: If you do not enter a name, DataAtlas will generate a nondescriptive name from the series CHECK1, CHECK2, ...

Comment

This field can contain the comment for a foreign key.

Required?: No.

Limit: 254 characters.

Check Condition

This field contains the text of the check constraint.

Required?: Yes.

Create

Click this button to create a new check constraint after you have:

1. Clicked the entry marked &lt.NEW> in the list of check constraints

2. Entered a name for the new check constraint

3. Entered the text of the check condition

Modify

Click this button to modify a check constraint after you have:

1. Clicked the entry in the list of check constraints

2. Updated the constraint's name, check condition, or both

Delete

Select an entry in the **List of check constraints** and click this button to delete the entry.

Options Page

On this page you can connect the table to or disconnect it from an existing table space and physical design.

Fields

- Table space
- INDEX IN
- LONG IN

List

- Physical design

Push Buttons

- Search
- Open
- Remove

Table Space

If you want to connect the table to a table space, select the adjacent **Search** push button and a table space from the resulting list. This field is filled in with the name of the table space you choose. Then open the table space (**Open** push button) and identify this table in the **Table Space** notebook.

Required?: No.

INDEX IN

In this field you can identify the table space in which indexes on the table will be created.

This field is enabled only if the **Table Space** field identifies a table space.

Required?: No.

LONG IN

In this field you can identify a table space in which the values of long columns - LONG VARCHAR, LONG VARGRAPHIC, LOB data types, or distinct types - will be stored.

This field is enabled only if the **Table Space** field identifies a table space.

Required?: No.

Physical Design

This is a list of physical designs to which the table belongs. You can connect the table to a new physical design by selecting the adjacent **Search** push button and a physical design from the resulting list. The name of the selected physical design will be added to the list on this page.

Required?: No.

Search

Click this button to see a list of table spaces or physical designs to which you can connect the table. The list of table spaces contains only those that belong to the relational system this table belongs to. (The name of the relational system is in the **System** field on the **General** page.)

Open

Click this button to open a **Table Space** notebook or a **Physical Design** notebook for a selected physical design.

Remove

Click this button to remove the contents of the **Table space** field or the name of a selected physical design from the **Physical design** list.

Indexes Page

This page enables you to add indexes to a table or to modify existing ones. The **Create** and **Open** push buttons open an index notebook for you to work with.

List

List of indexes

Push Buttons

Create
Open
Delete

List of Indexes

This list contains all the indexes that were defined for this table. If this a new table, the list will be empty. To create a new index, click **Create** . To modify or delete an existing index, select an entry in the list; then click **Open** or **Delete** , respectively.

Note: If the table is a new table and it has not been saved to the TeamConnection database, you will not be able to create an index for it. You must click **Save** first to save the table.

Create

Click this button to go create an index for this table.   This will open an Index notebook where you can define the index.

Note:

If the table is a new table, and it has not been saved to the TeamConnection database, you will not be able to create an index for it.   You must hit the Save button to save the table first.

Open

Click this button to modify an existing index after you have selected an index in the list of indexes . The Index notebook opens, enabling you to modify the attributes of the index.

Delete

Click this button to delete an existing index after you have selected on an index in the list of indexes . You will be prompted to confirm the delete.

General Page

This page enables you to enter and update general information about a DB2 UDB table. The **System** , **Schema** and **Database** fields cannot be modified for an existing table.

Fields

- System
- Schema
- Database
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the DB2 UDB table.

Required?: Yes.

DB2 UDB Table Space Notebook

This notebook enables you to view the tables and indexes in this table space.

Pages

- Contents
- Options
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Contents Page

This pages enables you to select a table or index and open its settings notebook.

List

Usage/Name

Push Button

Open

Options Page

On this page you can connect the table space to or disconnect it from an existing physical design.

Radio Buttons

- Table space type

List

- Physical designs

Push Buttons

- Search
- Open
- Remove

Table Space Type

Select the radio button in this box that specifies the type of table space this is. **REGULAR** means the table space stores any data except temporary tables. **LONG** means it stores long, or LOB, table columns. **TEMPORARY** means it stores temporary tables.

Default: **REGULAR** .

Physical Designs

This is a list of physical designs to which this table space belongs. You can connect the table space to a new physical design by selecting the adjacent **Search** push button and a physical design from the resulting list. The name of the selected physical design will be added to the list on this page.

Required?: No.

General Page

This page enables you to enter general information about an DB2 UDB table space.

Fields

- System
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the DB2 UDB table space.

Required?: Yes.

DB2 UDB View Notebook

This notebook enables you to display and modify a view's SQL definition and columns, and to comment on the view.

Pages

- Definition
- Options
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Definition Page

This pages enables you to:

⇑ See the SQL definition of a view
⇑ Add, modify, or delete view columns
⇑ Add a comment for a view and its columns

Fields

Fullselect
Comment (on the SQL statement)
Name
Comment (on a view column)

Check Box

Check option

List

Columns

Push Buttons

Add
Modify
Delete

Fullselect

This field contains the SQL query that defines the view. For a new view, you have to enter an SQL SELECT statement for the view.

Required?: Yes.

Comment (on the SQL Statement)

This field contains a comment on the SQL definition.

Name

In this field, you can enter alternative names for the column names that appear in the SQL definition of the view. When you select the Add push button, the name is added to the **Columns** list.

Column names in the **Columns** list must be in the same sequence as the column names in the SQL definition.

Required?: No.

Comment (on a View Column)

In this field, you can enter a comment on a view column.

Required?: No.

Check Option

Check this box if you want to enable the **Cascaded** and **Local** radio buttons. When **Cascaded** is selected, this view acquires as constraints the search conditions of any updatable view on which it depends. When **Local** is selected, this view's search condition becomes a constraint when an insert or update is performed on this view or on a dependent view.

Columns

This list contains the names of view columns and your comments about them.

Fill in the **Name** and **Comment** fields to add an entry to the list (with the **Add** push button) or to modify a selected entry (with the **Modify** push button).

You can delete an entry by selecting it and then selecting the **Delete** push button.

Add

Click this button to add a column name to the **Columns** list.

Modify

Click this button to modify a selected row in the **Columns** list.

Delete

Click this button to delete a selected row in the **Columns** list.

Options Page

On this page you can connect the view to or disconnect it from an existing physical design.

List

- Physical designs

Push Buttons

- Search
- Open
- Remove

General Page

This page enables you to enter general information about a DB2 UDB view. The **System**, **Schema**, and **Database** fields cannot be modified for an existing view.

Fields

- System
- Schema
- Database
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

System

This read-only field contains the name of the SQL server.

Required?: Yes.

Schema

This field contains the user ID under which the view was created. When a schema is selected by using the **Search** push button, both this field and the **System** field are initialized.

Required?: Yes.

Database

This field contains the name of the relational database.

Required?: Yes.

Name

This field contains the access name of the view.

Required?: Yes.

Variation

This field contains a qualifier that becomes part of the full name of the object in the TeamConnection database.   The qualifier can be useful in distinguishing between similar objects that have different uses.

Required?: No.

Default: Initialized from the profile.

Revision

This field contains a qualifier that becomes part of the full name of the object in the TeamConnection database. The qualifier can be useful in distinguishing between similar objects that have to be available at the same time.

Required?: No.

Default: Initialized from the profile.

Component

This field contains the name of the component to which the object belongs.

A component is a logical group of development data within a TeamConnection family. Identifying a component is a means of controlling access to development data.

Required?: No.

Limit: 31 characters.

Description

This field contains a text description of the object.

Required?: No.

Limit: 4096 characters.

History

This read-only field shows, by date and time, when the object was last modified.

DBD Notebook (All DBDs)

This notebook enables you to create and maintain an IMS database definition (DBD). Each DBD access method has a specific combination of DBD notebook pages and segment notebook pages. Only the parameters that are valid for each access method are presented.

Menu-Bar Choices

- DBD
- Window
- Help

Pages

- DBD
- DATASET (HDAM and HIDAM)
- DATASET (HIDAM)
- DATASET (HISAM)
- DATASET (HSAM and SHSAM)
- DATASET (INDEX)
- DATASET (MSDB)
- DATASET (SHISAM)
- AREA
- GSAM
- EXIT
- General

Push Buttons

- OK
- Save
- Cancel
- Help

DBD

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

Help

This choice displays a pull-down menu of the help information available with DataAtlas. The items on the pull-down menu are:

Help index. Gives you an alphabetic listing of the help topics for DataAtlas.

General help. Opens a window that explains the window or notebook you are using.

Using help. Opens a window that explains how to use the help function.

Help contents. Opens a table of contents of DataAtlas help topics.

Designer tasks. Takes you to the part of the help contents where the Designer tasks are explained.

Dictionary tasks. Takes you to the part of the help contents where the Dictionary tasks are explained.

Tutorial. Opens the DataAtlas online tutorial.

Product information. Opens a window that shows the DataAtlas logo and copyright information.

DBD Page

This page enables you to specify DBD parameters that are common to all access methods, and to manage the segments in the DBD.

Fields

- Access Method
- Version

Radio Buttons

- OSAM
- VSAM

Check Box

- Password

Table

- Segments

Push Buttons

- Create
- Open
- Delete
- Move Up
- Move Down

Access Method

This field contains a read-only copy of the access method on the General page.

Version

Enter a character string that identifies the DBD.   The exit routine is passed this string so it can determine the DBD version used to update the database. The string can be up to 255 bytes long.

OSAM

Select this button if you want to use OSAM as the operating system access method.

Default: Deselected.

VSAM

Select this button if you want to use VSAM as the operating system access method.

Default: Selected for GSAM, where the alternative to VSAM is BSAM, and for HDAM and HIDAM, where the alternative is OSAM. For INDEX, SHISAM, and HISAM DBDs, the button is selected but disabled, because the alternative to VSAM is ISAM, which is no longer supported.

Password

Check this box if you want the generated DBD to specify the PASSWD=YES parameter, which prevents accidental access to IMS databases by non-IMS programs.   DL/I uses the DBD name for this DBD as the VSAM password when opening any data set in the database.

This box is enabled only if the VSAM box is checked.

Segments

This table gives the name, parent, and length of each segment in the DBD.

Select a segment to be affected by the Open, Delete, Move up, or Move down button.

Create

Click this button to add a new segment to the DBD.

Open

Click this button to open a selected segment.

When you open a segment, the properties notebook for the segment is displayed.

Delete

Click this button to remove a selected segment from the DBD.

Move Up

Click this button to move a selected segment up in the table.

Move Down

Click this button to move a selected segment down in the table.

DATASET Page (HDAM)

This page specifies the parameters that are specific to a HDAM DBD.   You can define a new DBD by filling in the fields at the bottom of the page for each data set in the DBD.

Fields

- Randomizing module name
  - Anchor points
  - Block number
  - Root bytes
- Label
- DDName
- Device, Model
- Size
- Scan
- Free Block Frequency
- Free Space Percentage

Table

- Data Sets

Push Buttons

- Create
- Modify
- Delete

Radio Buttons

- Search Algorithm

Anchor Points

This field contains the number of root anchor points you want in each control interval or block in the root-addressable area. Use an unsigned decimal number.

Limit: 255 or less. Values from 1 to 5 are typical.

Block Number

This field contains the maximum relative block number that you will allow the randomizing module to produce. This value determines the number of control intervals or blocks. The number must be an unsigned decimal integer.

Limit: 16777215.

Default: None.

Root Bytes

This field contains the maximum number of bytes of a database record that can be stored into the root-addressable area in a series of inserts unbroken by a call to another database record. If the Block Number field is omitted, the value in this field is ignored.

Limit: 16777215.

Default: None.

Label

This field contains a label for the DATASET statement when one is generated. All segments (defined by SEGM statements) that follow DATASET statements with a given label are placed in the same data set group.   The data set group is defined by the first DATASET statement with the label.

The label must be alphanumeric and conform to the requirements for labels on MVS assembler language statements.

Required?: No.

DDName

This field contains the ddname of the primary data set in this data set group. The name will also appear in the DD1 column of the Data Sets table.

Required?: Yes.

Limit: 8 alphanumeric characters.

Device, Model

This scrollable field lets you specify the storage device and, where appropriate, the model on which the data set will be stored.

Required?: Yes.

Size

This field contains a size in bytes that is used to override DBDGEN's computation of control interval or block size. If the value differs from the control interval size defined to VSAM, DL/I uses the value defined to VSAM.

For VSAM data sets, values less than 8192 must be multiples of 512. Values from 8192 to 30720 must be multiples of 2048.

For OSAM data sets, the maximum block size is 32K.

Required?: No.

Scan

This field contains the number of cylinders to be scanned when searching for storage space during segment insertion operations. The number must be a decimal integer.

Required?: No.

Default: 3. Typical values are 0 to 5.

Limit: 255.

Free Block Frequency

This field specifies how you want to distribute free space. The number it contains, ***n*** , specifies that every ***n*** th control interval or block in the data set group will be left as free space during the loading or reorganization of the database.

Required?: No.

Default: 0.

Limits: 0 to 100, excluding 1.

Free Space Percentage

This field contains the minimum percentage of each control interval or block in the data set group that you want to reserve for free space.

Required?: No.

Default: 0.

Limits: 0 to 99.

Data Sets

This table reflects the specifications for the data sets in the DBD, as they were entered in the fields on the **DATASET** page.

Select a data set to be affected by the **Modify** or **Delete** button.

To specify the data set in which you want to locate a given segment, go to the **DBD** page, select the segment, open it (which opens the **Segment** notebook), and go to the **SEGM** page.

Create

Click this button to add a new data set.

Modify

Click this button to modify the selected data set.

Delete

Click this button to remove the selected data set.

Search Algorithm

Click a button that specifies the type of HD search algorithm you want IMS to use to insert a segment.

0: IMS chooses whether to use option **1** or **2** . For Versions 3, 4, and 5 of IMS/ESA, IMS chooses **2** .

1: IMS uses the algorithm that does **not** search for the second-most desirable block or control interval.

2: IMS uses the algorithm that searches for the second-most desirable block or control interval.

none: The same as clicking on 0.

DATASET Page (HIDAM)

This page enables you to specify the attributes of a HIDAM DBD. You can define a new DBD by filling in the fields at the bottom of the page for each data set in the DBD.

Fields

- Primary index DBD
- Segment
- Label
- DDName
- Device, Model
- Size
- Scan
- Free Block Frequency
- Free Space Percentage

Table

- Data Sets

Push Buttons

- Search
- Create
- Modify
- Delete

Radio Buttons

- Search Algorithm

Primary Index DBD

Specify the name of the index DBD in this field.   It is used to generate the LCHILD statement on the root segment.

Segment

This is a read-only field that shows the name of the segment in the index DBD.

Search

Click this button to search for an index DBD in the TeamConnection database.

DATASET Page (HISAM)

This page specifies the parameters for a HISAM DBD.   Its fields are in two groups: one for the primary data set, one for the overflow data set.

Fields

- DDName
- Device, Model
- Size
- Record Length
- Blocking Factor

DDName

For the primary data set, this field contains the value to be used for the DD1 parameter; that is, the ddname of the primary data set.

For the overflow data set, this field contains the value to be used for the OVFLW parameter; that is, the ddname of the overflow data set.

Required?: Yes.

Limit: 8 Alphanumeric characters.

Size

These fields specify the control interval or block size of the primary and overflow data sets.

For VSAM data sets, values less than 8192 must be multiples of 512.   Values from 8192 to 30720 must be multiples of 2048.

Record Length

These fields specify the data management logical record lengths to be used.

Required?: No.

Blocking Factor

These fields specify the blocking factor you want to use for the primary and overflow data sets.

The blocking factor, multiplied by the record length, determines the block size of the data set.

If you specify a value in the Size field, any value you specify here is ignored.

Required?: No.

DATASET Page (HSAM and SHSAM)

This page enables you to specify parameters for an HSAM or SHSAM DBD.   Its fields are in two groups: one for the input data set, one for the output data set.

Fields

- DDName
- Device, Model
- Record Length
- Blocking Factor

DDName

For the input data set, this field contains the value to be used in the DD1 parameter; that is, the ddname of the input data set.   It must be a 1- to 8-character alphanumeric name.

Required?: Yes.

For the output data set, this field contains the value to be used for the DD2 parameter; that is, the ddname of the output data set.   It also must be a 1- to 8-character alphanumeric name.

Required?: Yes.

Record Length

These fields specify the data management logical record lengths to be used.

Required?: No.

Blocking Factor

These fields specify the blocking factor you want to use for the input and output data sets.

The blocking factor, multiplied by the record length, determines the block size of the data set.

Required?: No.

DATASET Page (INDEX)

This page enables you to describe an index DBD and its target DBD.

Fields

- Index Target DBD
- Segment
- Field Name
- DDName
- Device, Model
- Size
- Record Length
- Blocking Factor

Check Boxes

- Primary Index
- Protect
- DOS Compatible

Push Buttons

- Search
- Details

Index Target DBD

This field contains the name of the DBD for the target database of the index. Select the **Search** push button and either select a name from the list that's displayed or type in a name.

Required?: Yes.

Segment

This field identifies the segment type in the target database that is being indexed.   For a primary index, this is a read-only field that DataAtlas fills in when you have identified the target database.   For a secondary index, you select the segment type from the list connected to this field.   DataAtlas creates the list when you have identified the target database.

Required?: Yes.

Field Name

This field identifies:

⇑ For a primary index, the sequence field of a HIDAM root segment type in the target database.   DataAtlas fills in the field when you have identified the target database.
⇑ For a secondary index, the indexed data field of an index target segment. In this case, you have to enter the name of the field.

Required?: Yes.

Size

For the primary and overflow data sets, these fields specify the control interval or block size.

These fields are mutually exclusive with the **Blocking Factor** fields.

Default: If you do not specify values, DBDGEN generates them.

Record Length

These fields specify the logical record lengths in the primary and overflow data sets.

Default: If you do not specify values, DBDGEN generates them.

Blocking Factor

These fields specify the blocking factors for the primary and overflow data sets.

These fields are mutually exclusive with the **Size** fields.

Default: If you do not specify values, DBDGEN generates them.

Primary Index

Check this box is checked if the target database will be a HIDAM database; otherwise; leave it unchecked.

Default: If the target database has been identified as a HIDAM database, this box will be checked. If it has been identified as a non-HIDAM database, this box will be disabled.

Protect

This box applies only to secondary index databases. Check it to prevent an application program from **replacing** any field in an index pointer segment except for fields in the user data part of the segment. An application can still **delete** index target segment pointers when this option is checked.

Default: Checked for secondary index databases.

DOS Compatible

Check this box if the index was created using DLI/DOS.

Details

Click this button to enter detailed information for a secondary index -- information that corresponds to the specifications on an XDFLD statement.

DATASET Page (MSDB)

This page enables you to specify the REL parameter in the DATASET statement for an MSDB DBD.

Field

Field Name

Combination Box

Database Type

Field Name

This field lets you specify the 1- to 8-character alphanumeric name of a search field.   The name must be different from any other field name defined in a FIELD statement.

Required?: No.

Database Type

This box lets you specify whether an MSDB is terminal-related (**FIXED** or **DYNAMIC**) or non-terminal-related (**NO** or **TERM**).

Each segment in a terminal-related MSDB is assigned to a different LTERM. The LTERM name is the segment key but is not contained in the segment. Each LTERM owns no more than one segment per MSDB, and only the owner can alter a segment.

NO specifies a non-terminal-related MSDB without terminal-related keys. The key and sequence field are part of the segment. The Field Name field is disabled.

TERM specifies a non-terminal-related MSDB with terminal-related keys. The key is the LTERM name (not part of the segment); there is no sequence field.

FIXED specifies a terminal-related fixed MSDB. The LTERM name is the segment key. Segment updates are allowed. Segment insertions and deletions are not allowed.

DYNAMIC specifies a terminal-related dynamic MSDB. The LTERM name is the segment key. No more than one insertion or deletion can be made to the same MSDB from a single LTERM within one sync processing interval.

Required?: Yes.

DATASET Page (SHISAM)

This page enables you to specify parameters for a SHISAM DBD.

Fields

- DDName
- Device, Model
- Size
- Record Length
- Blocking Factor

DDName

This field contains the ddname of the primary data set -- the value used in the DD1 parameter.   The ddname must be a 1- to 8-character alphanumeric name.

Required?: Yes.

Size

This field specifies the control interval or block size of the primary data set.

Values less than 8192 must be multiples of 512.   Values from 8192 to 30720 must be multiples of 2048.

Required?: No.

Record Length

This field specifies the data management logical record length to be used.

For simple HISAM databases, the logical record length must be the same as the segment length specified.

Required?: No.

Blocking Factor

This field specifies the blocking factor you want to use for the primary data set.

The blocking factor, multiplied by the record length, determines the block size of the data set.

If you specify a value in the Size field, any value you specify here is ignored.

Required?: No.

AREA Page

This page specifies the parameters that are specific to a DEDB DBD.   You can define a new DEDB by filling in the fields at the bottom of the page for each DEDB area.

Fields

- Randomizing module name
- Randomizer stage
- DDName
- Device, Model
- Units of Work
  - Overflow
- Size
- Root
  - Overflow

Check Box

- Extended call interface

Table

- Areas

Push Buttons

- Add
- Modify
- Delete

Randomizing Module Name

This field contains a 1- to 8-character alphanumeric name of a user-supplied randomizing module used to store and access segments.

Required?: Yes.

Randomizer Stage

If you're running IMS Version 6 or higher, use this option to specify the type of randomizer to be used.

1	designates a 1-stage randomizer; it randomizes across the entire database. If this attribute is changed in the generated DBD, the entire database must be reorganized.
2	designates a 2-stage randomizer; it randomizes to an area first, and then to a root anchor point within the area. If this attribute is changed in the generated DBD, only the affected areas have to be reorganized.
0	means you want the IMS system to determine which type of randomizer to use.
Required?:	No.
Default?:	0.

DDName

This field contains the ddname of the 1- to 8-character alphanumeric name of a defined area.   The name will also appear in the DD1 column of the Areas table.

Required?: Yes.

Device, Model

When defining an area, this scrollable field lets you specify the storage device and, when appropriate, the model on which an area will be stored.

Required?: Yes.

Units of Work

When defining an area, enter the number of control intervals in a unit of work here.

Limit: 2 to 32767.

Required?: Yes.

Overflow (UOW)

In this field, specify the number of control intervals in the overflow section of a unit of work.

Limits: At least 1. At most 1 less than the value in the Units of Work field.

Size

In this field, specify the size of the control interval.   It can be 512 bytes, 1KB, 2KB, 4KB, and multiples of 4KB up to 28KB.

Required?: Yes.

Restriction: With a 2319 device, 4KB cannot be specified.

Root

In this field, specify in units of work the total space allocated to the root-addressable part of the area and to the area reserved for independent overflow.   The rest of the VSAM data set is reserved for sequential dependent data.

Required?: Yes.

Limits: 3 to 32766.   The allocated space cannot be larger than the amount of space actually in the VSAM data set.

Overflow (Root)

In this field, specify in units of work the space reserved for independent overflow.

Limits: At least 1. At most 1 less than the value in the Root field.

Extended Call Interface

If you're running IMS Version 6 or higher, check this box to include an XCI subparameter when you generate this DBD. The subparameter causes a user-written initialization routine to be invoked when the randomizer is loaded and a user-written termination routine to be invoked when it's unloaded. This option is useful for loading tables the randomizer uses.

Adding the XCI subparameter is treated as a database-level change for online change purposes.

This check box is disabled if the value in the **Randomizer stage** field is 0.

Areas

This table contains information about each of the areas in the DEDB.

Select an area to be affected by the Modify, Delete, Move up, or Move down button.

Add

Click this button to add a new area to the DEDB.

Modify

Click this button to modify the selected area.

Delete

Click this button to remove the selected area.

Move Up

Click this button to move the selected area up in the list of areas.

Move Down

Click this button to move the selected area down in the list of areas.

GSAM Page

This page enables you to specify DBD and DATASET parameters for a GSAM DBD.

Fields

- Access Method
- Version
- DDName (Input Data Set)
- DDName (Output Data Set)
- Maximum (Record Length)
- Minimum (Record Length)
- Size
- Blocking Factor

Radio Buttons

- VSAM
- BSAM

Check Box

- Password

List

- Record Format

Access Method

This read-only field is a copy of the **Access Method** field on the General Page.

Version

This field contains a character string that is used to identify the DBD.

Limit: 255 bytes.

Required?: No.

Default: A 13-character time stamp created by DBDGEN.

DDName (Input Data Set)

This field contains the name of the input data set, DD1.

Required?: Yes.

Limit: 8 alphanumeric characters.

DDName (Output Data Set)

This field contains the name of the output data set, DD2.

Required?: No. If you don't specify a name, the input data set is assumed to be the output data set as well.

Limit: 8 alphanumeric characters.

Maximum (Record Length)

This field specifies the size of a fixed-length record or the maximum size of a variable-length or undefined record.

Default: If you do not enter a value, DBDGEN generates one.

Minimum (Record Length)

This field specifies the minimum size of a variable-length or undefined record.

Default: If you do not enter a value, DBDGEN generates one.

Size

This field contains the block size for the input/output data set.

Required?: No.

Default: If you do not specify a value, DBDGEN generates one.

Blocking Factor

This field contains the blocking factor for the input/output data set.

Required?: No.

Default: If you do not specify a value, DBDGEN generates one.

VSAM

Select this button to specify VSAM as the operating system access method.

Default: Selected.

BSAM

Select this button to specify BSAM as the operating system access method.

Default: Deselected.

Record Format

Select from the list a format for the records in the data set. The keywords in the list have the following meanings:

F	Fixed-length records
FB	Fixed-length, blocked records
V	Variable-length records
VB	Variable-length, blocked records
U	Undefined-length records

EXIT Page

This page enables you to identify Data Capture exit routines and select data options for them.

The Data Capture exit routines identified here apply to all the segments in the physical database. To apply these routines to particular segments, use the **EXIT** page in the **Segment** notebook.

Table

Data Capture Exits

Field

Capture Routine

Check Boxes

Data
Key
Path
Log
Cascade

Push Buttons

Add
Modify
Delete

Data Capture Exits

Each row of this table records an entry in the **Capture Routine** field and your check-box selections for that entry. The **Add** push button adds a row to the table. **Modify** lets you change a selected row. **Delete** lets you delete a selected row.

The order of the entries in this table indicates the order in which the routines are called to process a segment.

Capture Routine

This field contains the name of the exit routine that processes the data. You can specify an asterisk (*) instead of a name to indicate that you want logging only.

Limit: 8 alphanumeric characters.

Data

Check this box to specify that the physical segment data is passed to the exit routine for updating.

If this box is checked and a Segment/Compression exit routine is also used, the data passed to the Data Capture exit routine is expanded data.

Default: Checked.

Key

Check this box if you want the physical concatenated key to be passed to the exit routine.   This key identifies the physical segment updated by the application.

Default: Checked.

Path

Check this box if the exit routine requires data from segments in the physical root's hierarchical path.

Default: Unchecked.

Log

Check this box if you want IMS data control blocks and data to be written to the IMS system log.

Default: Checked only if an asterisk (*) is specified in place of an exit routine name.

Cascade

Check this box if you want the exit routine to be called for a segment that IMS deleted because the application deleted its parent.

Default: Checked.

The **Data**, **Key**, and **Path** check boxes that you see under the **Cascade** check box apply to segments deleted when the application deletes its parent. The meaning and defaults for these check boxes are the same as for the check boxes that don't apply to cascading deletions.

Add

Click this button to add the name of a Data Capture exit routine and its data options to the **Data Capture Exits** table.

Modify

Click this button to replace a selected row in the **Data Capture Exits** table with a different exit routine name and check-box selections.

Delete

Click this button to delete a selected row from the **Data Capture Exits** table.

General Page

This page enables you to:

⇑ Name or rename a DBD and specify its access method
⇑ Enter or change the DBD name's qualifiers
⇑ Enter or change the component the DBD belongs to
⇑ Enter or change text that describes the DBD

Fields

- Access method
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Set Access Method
- Search

Access Method

This field identifies the DL/I access method to be used for this DBD.

For a new DBD, select an access method from the list below this field. When you click **Set Access Method** , the access method you have chosen is locked in.

Here is what each of the items in the list means:

DEDB. The DBD describes a data entry database (DEDB).

GSAM. The Generalized sequential Access Method (GSAM) will be used for the database.

HDAM. The Hierarchical Direct Access Method (HDAM) will be used for the database. OSAM or VSAM can be specified as the operating system access method. VSAM is the default.

HIDAM. The Hierarchical Indexed Direct Access Method (HIDAM) will be used for the database. OSAM or VSAM can be specified as the operating system access method. VSAM is the default.

HISAM. The Hierarchical Indexed Sequential Access Method (HISAM) will be used for the database.

HSAM. The Hierarchical Sequential Access Method (HSAM) will be used for the database.

INDEX. The DBD is for a primary index to occurrences of the root segment type or for a secondary index to a segment type in a HDAM, HIDAM, or HISAM database.

LOGICAL. The DBD describes a logical database, a database composed of one or more physical databases. A logical DBD generation is meaningful only if a physical DBD generation exists that defines segment types referenced by SEGM statements in the logical DBD generation.

MSDB. The DBD describes a main storage database (MSDB).

SHISAM. This is a simple HISAM database; it contains only one fixed-length segment type. No prefix is required in occurrences of the segment type to enable IMS to process the database.

SHSAM. This is a simple HSAM database; it contains only one fixed-length segment type. No prefix is required in occurrences of the segment type to enable IMS to process the database.

Name

This field contains the access name of the DBD being described.

The name can be the same as the name specified in the DD1 column of the DATASET page.

Required?: Yes.

Limit: 8 alphanumeric characters.

Component

This field contains the name of a component.

A component is a logical group of development data within a TeamConnection family. Identifying a component is a means of controlling access to development data.

Use the **Search** push button to see a list of names that can be entered here.

Required?: No.

Set Access Method

Click this button to lock in the access method you have chosen for this DBD.

OK

Click this button to save any changes you made and close the window or notebook.

Save

Click this button to save any changes you made and leave the window or notebook open.

Recommendation: Use this button intermittently to save your work when you are in a long updating session.

Cancel

Click this button to close the window or notebook without saving any changes you may have made.

Help

Click this button to see an explanation of the window or notebook you are viewing.

Delete Confirmation Window

This window enables you to review the deletion choices you made. You can decide to continue the deletion process or cancel it.

Recommendation: If you're not sure how deleting an object would affect other objects in the TeamConnection database, run a supplied query to learn more about the object's relationships to other objects.

Push Buttons

- Delete
- Cancel
- Help

Delete

Click this button to delete the listed objects from the TeamConnection database.

Cancel

Click this button to stop the deletion process. This window will close, and you will return to the **Delete** page of the **Delete** notebook.

Delete Notebook

This notebook enables you to delete objects from the TeamConnection database.

Pages

Delete
TeamConnection

Push Buttons

Delete
Cancel
Help

Delete Page

This page enables you to delete from the TeamConnection database the objects you selected **and** the objects related to them.

The top table on this page, **Objects to be deleted**, lists the objects you selected from the workfolder. The top entry is initially highlighted -- it is selected for deletion. Related objects are listed in the bottom table, **Related objects to be deleted**. They too are selected for deletion. If you decide not to delete a related object, deselect it.

If you highlight another entry in the top table, DataAtlas does two things:

⇑ It saves the deletion selections you have already made.

⇑ It displays in the bottom table the objects related to the entry you highlighted. Again, the related objects are selected for deletion.

Continue the process of displaying related objects until you have deselected any that you want to keep. Then select the **Delete** push button**.** All the objects in the top table and all the related objects that remained selected are deleted from the TeamConnection database, unless you checked the Confirm on Delete box.

Recommendation: If you're not sure how deleting an object would affect other objects in the TeamConnection database, run a supplied query to learn more about the object's relationships to other objects.

Confirm on Delete

Check this box if you want to see the **Delete Confirmation** window when you select the **Delete** push button. The window lets you review your deletion choices before continuing the deletion process.

Default: Unchecked.

TeamConnection Page

This page enables you to:

⇑ Include remarks when you delete original objects from the TeamConnection database

⇑ Extend your delete request to other releases and work areas

⇑ Break a common link

Field

Remarks for Checkin

Check Box

Break common link

Remarks for Checkin

This field contains a text comment about the deleted object.

Required?: No.

Break Common Link

Check this box if, in deleting the selected objects from your work area, you do not want to affect the objects in a linked work area.

Default: Unchecked.

Delete

Click this button to see a confirmation window that lists all the objects you have selected for deletion. You can actually delete the objects by clicking **Continue** on that window.

Export to File Window

This window enables you to export an SQL query report to a file. The report is exported in ASCII delimited text.

Field

Save as filename

Lists

Type of file
Drive
Directory
File

Push Buttons

OK
Cancel
Help

Export Selected to File Window

This window enables you to export selected rows from an SQL query report to a file. The rows are exported in ASCII delimited text.

Field

Save as filename

Lists

Type of file
Drive
Directory
File

Push Buttons

OK
Cancel
Help

Field Notebook

This notebook enables you to name an IMS field, write a text description of it, and define it.

Fields are referred to by PSBs to specify sensitivity to the fields or by an application program in a DL/I call segment search argument.

A unique sequence field must be defined for the root segment types of HISAM, HIDAM, primary HIDAM INDEX, SHISAM, DEDB, and non-terminal-related MSDB databases.   Root segment types in an HDAM database do not need a key field defined; if one is defined, it doesn't have to be unique.

Menu-Bar Choices

- Field
- Window
- Help

Pages

- Field
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Field

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

Field Page

This page enables you to define an IMS field for a given segment type.

Fields

- Bytes
- Start position
- Shareable data element name

Lists

- Data item
- Data type

Check Boxes

- Sequence
- Multiple
- Use existing shareable data element

Push Buttons

- Search
- Open

Bytes

This field specifies the length of the IMS field in bytes.

Required?: Yes.

Limits:

⇑ 255, for a field that isn't system-related.

⇑ The length of the concatenated key of the index source segment, for a system-related field that is all or part of a concatenated key of an index source segment.

⇑ 4, for an /SX system-related field.

⇑ 240, for the sequence field of an MSDB segment.

⇑ The value of minbytes specified for the segment, for a sequence field of a DEDB segment.

Start Position

This field specifies the starting position of the IMS field in terms of bytes relative to the beginning of the segment.

Required?: Yes.

Shareable Data Element Name

If you search for and select a shareable data element , DataAtlas puts the name of the selected element in this field.

Data Item

If this field's segment is connected to a shareable data structure, this list is initialized with the names of the data items in that structure. You can select the data item that should be connected to this field.

Data Type

Select from this list a character that indicates the type of data contained in this IMS field. The characters have these meanings:

C. Alphanumeric data or a combination of data types.

F. Binary fullword data.

H. Binary halfword data.

P. Packed decimal data.

X. Hexadecimal data.

Required?: No.

Default: **C** .

Restriction: F and H are valid only for MSDB databases.

Sequence

Check this box if the IMS field is a sequence field.

Default: Unchecked.

Multiple

Check this box is this is multiple sequence field.

This box is available only when the **Sequence** box is checked.

Default: Unchecked.

Use existing shareable data element

Check this box if you want to use a shareable data element to define the field. When you check this box, the **Search** and **Open** push buttons are enabled.

Default: Unchecked.

Search

Click this button to get a list of shareable data elements in the TeamConnection database. When you select a name from the list, it's put in the **Shareable data element name** field.

Open

Click this button to open the properties notebook for the shareable data element named in the **Shareable data element name** field.

General Page

This page enables you to name the IMS field and write a description of it.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the IMS field.

Required?: Yes.

Variation

This field contains the name of a variation.

A variation name becomes part of the full name of an object in the TeamConnection database. It can be useful in distinguishing between similar objects that have different uses.

If an object is related to a higher-level object, it takes the variation name, if any, of the higher-level object. So, for example, an object for an IMS field or segment will automatically be assigned the variation name given to the higher-level DBD when it was defined.

Required?: No.

Revision

This field contains the name of a revision.

A revision is a qualifier that becomes part of the full name of an object in the TeamConnection database.   This qualifier can be useful in distinguishing between similar objects that have to be available at the same time.

If an object is related to a higher-level object, it takes the revision name, if any, of the higher-level object.   So, for example, an object for an IMS field or segment will automatically be assigned the revision name given to the higher-level DBD when it was defined.

Required?: No.

Font Window

To choose a font:

1. Click the arrow to the right of the **Name** field to display a list of font names. Select a font name from the list.

2. Click the arrow to the right of the **Size** field to display the list of font sizes. Select a font size from the list.

3. Click the arrow to the right of the **Style** field to display the list of font styles. Select a font style from the list.

4. Check a box in the **Emphasis** group if you want the text emphasized in the indicated way.

The **Display** check box is selected by default. Printer font selection is not enabled.

Lists

- Name
- Size
- Style

Check boxes

- Emphasis

Push Buttons

- OK
- Save
- Cancel
- Help

Name

The list shows the names of the fonts available on your system. When you select a font, it is displayed in the **Sample** area.

Size

The list contains all the sizes in which the selected font can be displayed. When you select a size, it is displayed in the **Sample** area.

Style

The list shows the type styles you can choose for the selected font. When you select a style, it is displayed in the **Sample** area.

Emphasis

The check boxes shows the kinds of emphasis you can choose for the selected font. When you make a choice, it is displayed in the **Sample** area.

Ok

Click **Ok** to display the text in the font specifications you selected.

Save

Click **Save** to display the text in the font specifications you selected.

When you use **Save**, the **Font** window stays open.

Generate Output File - Search Window

This window enables you to select a file that will contain output from a DB2 or Oracle generate operation.

Field

- Open filename

Lists

- Type of file
- Drive
- Directory
- File

Push Buttons

- OK
- Cancel
- Help

Open Filename

This field identifies the file that will receive the output of a generate operation. You can fill in the field by typing into it directly or by clicking an entry in the **File** list.

When you click **OK** , the name in this field is entered into the **Output file** field in the **Generate to File** notebook.

Required?: Yes.

Type of File

Select from the drop-down list the file type to be used. The type you choose will delimit the file names displayed in the **File** list.

Required?: Yes.

Default: All the files in the directory selected from the **Directory** list.

Drive

Select from the drop-down list the drive that contains the file you want.

Required?: Yes.

Default: The drive of the current directory.

File

Click a file name in this list and it's entered into the **Open filename** field. If you double-click a file name, it's entered directly into the **Output file** field in the **Generate to File** notebook, and this window closes.

Generate to File Notebook

This notebook enables you to generate files that contain the definitions of objects stored in the TeamConnection database. You'll see only the pages related to the objects you selected.

Pages

- DBD/PSB
- DB2
- Oracle
- Physical Design
- Included Source
- Compile Options

Push Buttons

- Generate
- Cancel
- Help

Compile Options Page

DataAtlas validates generated COBOL and PL/I output by compiling it. On this page, specify the options you want the compiler to use.

You can also specify:

⇑ The level at which the generated source is to start
⇑ The increment to be used from level to level
⇑ Any SPECIAL NAMES clauses (COBOL only)
⇑ A prefix and suffix to be added to a data name (COBOL only)

Check Box

Use command file not compiler options

Fields

Compile options
Command file
SPECIAL NAMES (COBOL only)
Prefix (COBOL only)
Suffix (COBOL only)

Push Button

Search

Spin Buttons

Start level
Increment

Compile Options

This field contains the compiler options you want to use in the compilation phase of generating an included source definition.   Alternatively, you can put compiler options in a command file.

Required?: No.

Default: Initialized from the profile.

Command File

This field identifies a file that invokes the compiler and passes it compiler options. Alternatively, you can specify compiler options in the **Compile options** field.

The **Use command file not compiler options** box enables the **Command file** field.

Required?: No.

Default: Initialized from the profile.

Note: The path `&lt.DataAtlasInstallPath>\lang\en_us\samples\cobol` contains a sample COBOL command file named `ewslpgm.cmd`. The path `&lt.DataAtlasInstallPath>\lang\en_us\samples\pli` contains a sample PL/I command file named `ewslppg.cmd`. `&lt.DataAtlasInstallPath>` is the root directory where DataAtlas is installed.

Start level

Select the level at which the generated source will start.

Default: Initialized from the profile.

Increment

Select the level-to-level increment you want in the generated source.

Default: Initialized from the profile.

SPECIAL NAMES (COBOL only)

Enter in this field any of the clauses of the SPECIAL NAMES paragraph that you want to give to the compiler. This field is enabled only if the **Generate source as** field on the **Included Source** page specifies 'COBOL'.

Required?: No.

Default: Initialized from the profile.

Prefix (COBOL only)

This field can contain a string to add to the beginning of a COBOL data name. It is enabled only if the **Generate source as** field on the **Included Source** page specifies 'COBOL'.

Required?: No.

Default: Initialized from the profile.

Suffix (COBOL only)

This field can contain a string to add to the end of a COBOL data name. It is enabled only if the **Generate source as** field on the **Included Source** page specifies 'COBOL'.

Required?: No.

Default: Initialized from the profile.

DB2 Page

This page enables you to specify a file that will contain the generated output for the objects you selected.

Lists

- Objects selected
- Type

Fields

- Output file
- Comparison database

Check Boxes

- Append to existing file
- Generate object descriptions for DataGuide
- Convert names to uppercase

Radio Buttons

- Create
- Alter
- Drop
- Drop/Create

Push Buttons

- Search (Output file)
- Search (Database alias)

Objects Selected

This list shows the fully qualified names of the DB2 objects you selected.

Type

This list shows the types that correspond to the objects in the **Objects selected** list.

Output File

This field names the file to which DataAtlas generates output for DB2 objects. Specify a full path for the file unless you want it to be in the directory where DataAtlas is running. If so, the file name alone is sufficient.

Required?: Yes.

Default: Initialized from the profile.

Comparison database

This field contains the database alias you selected after using the adjacent **Search** push button. DataAtlas uses it to connect to the database catalog and compare the cataloged definition with the definition in the TeamConnection database.

Required?: Only if the **Alter** radio button is selected.

Append to Existing File

Check this box to append the generated output to the contents of an existing file.   If you do not check this box and the file exists, DataAtlas will overwrite its contents.

Default: Initialized from the profile.

Convert Names to Uppercase

Check this box if you want DataAtlas to convert all the variable names in the generated DDL to uppercase.

Default: Unchecked.

Search (Output File)

Click this button to search for and select an existing file that will contain the generated output.

Search (Database Alias)

Click this button if you selected the **Alter** radio button. It opens a list of database aliases. DataAtlas uses the one you choose to determine how the object's definition has changed.

Oracle Page

This page enables you to specify a file that will contain the generated output for the objects you selected.

Lists

- Objects selected
- Type

Fields

- Output file

Check Boxes

- Append to existing file
- Generate object descriptions for DataGuide
- Generate OR REPLACE clause for views
- Convert names to uppercase

Radio Buttons

- Create
- Drop
- Drop/Create

Push Button

- Search

Objects Selected

This list shows the fully qualified names of the Oracle objects you selected.

Output File

This field names the file to which DataAtlas generates output for Oracle objects.   Specify a full path for the file unless you want it to be in the directory where DataAtlas is running.   If so, the file name alone is sufficient.

Required?: Yes.

Default: Initialized from the profile.

Generate OR REPLACE Clause for Views

Check this box if you want an OR REPLACE clause to be generated. Valid only if you are generating an Oracle view.

Required?: No.

Search

Click this button to search for and select an existing file that will contain the generated output.

Physical Design Page

This page enables you to specify a file that will contain the generated output of the physical designs you selected. The generated output will consist of the DDL for all the objects in each of the selected physical designs.

Lists

- Objects selected
- Type
- Output file

Field

- Output directory

Check Box

- Append to existing file
- Generate object descriptions for DataGuide
- Convert names to uppercase

Radio Buttons

- Create
- Drop
- Drop/Create

Push Button

- Search

Objects Selected

This list shows the fully qualified names of the physical designs you selected.

Type

This list identifies the physical designs as DB2/390, DB2 UDB, or Oracle physical designs.

Output File

This list shows the generated file names that correspond to the physical designs in the **Objects selected** list.

To change the default name of an output file:

1. Hold down the ALT key and click the default name.
2. Type in the new file name.
3. Click anywhere in the **Output File** list.

Default: Initialized from the profile.

Output Directory

This field identifies a fully qualified directory; that is, a drive and a directory name.

Default: Initialized from the profile.

DBD/PSB Page

This page enables you to specify the directory where output files will be created for the objects you selected.

Lists

- Objects selected
- Output file

Field

- Output directory

Push Button

- Search

Check Box

- Generate object descriptions for DataGuide

Radio Buttons

- Never, On errors, Always

Objects Selected

This list shows the fully qualified access name of the objects you selected.

Output File

This list shows the generated file names that correspond to objects in the **Objects selected** list.

To change the default name of an output file:

1. Hold down the ALT key and click the default name.
2. Type in the new file name.
3. Click anywhere in the **Output File** list.

Output Directory

This field identifies a fully qualified directory; that is, a drive and a directory name.

Required?: Yes.

Default: Initialized from the profile.

Search

Click this button to search for and select an existing directory that will contain the generated output.

Never, On Errors, Always

Select the button that tells DataAtlas when to produce a report of the generate process - never, when an error occurs, or always.

Default: Initialized from the profile.

Included Source Page

This page enables you to specify the directory where output files will be created for the objects you selected.

Lists

Objects selected
Output file

Fields

Output directory
Generate source as

Push Button

Search

Radio Buttons

Never, On errors, Always

Generate Source As

This field indicates the type of source statements to be generated. You can select 'COBOL' or 'PL/I'.

Required?: Yes.

Default: Initialized from the profile.

Generate

Click this button to generate files containing the definitions of the objects selected in the **Objects selected** list.

Import File - Search Window

This window enables you to select a file and write its contents on the **Query** page of the **SQL Query** notebook.

Field

Open filename

Lists

Type of file
Drive
Directory
File

Push Buttons

OK
Cancel
Help

Open Filename

This field identifies the file whose contents you want to import. You can fill in the field by typing into it directly or by clicking an entry in the **File** list.

When you click **OK**, the contents are written on the **Query** page of the **SQL Query** notebook you came from. The fields on the notebook's **General** page, which can contain descriptive information about a query, are not affected.

Required?: Yes.

File

Click a file name in this list and it's entered into the **Open filename** field. Double-click a file name and the contents are written on the **Query** page of the **SQL Query** notebook you came from. The fields on the notebook's **General** page, which can contain descriptive information about a query, are not affected. Double-clicking a file name also closes this window.

IMS-COBOL-PL/I Search Window

This window enables you to select workstation files that contain DBD, PSB, COBOL, or PL/I data definitions you want to populate.

Field

- File name

Lists

- Drive
- Directory
- File

Push Buttons

- Continue
- Cancel
- Help

File Name

In this field you can enter a file-name criterion that will delimit the file names displayed in the **File** list. After you select from the **File** list, this field will contain the first or only file name that you selected.

Required?: Yes.

Default: ' *****' (which does not delimit the list of file names).

Drive

This list identifies the drives you can select from.

Required?: Yes.

Default: The drive specified on the **IMS** , **COBOL** , or **PL/I** page of your profile.

Directory

This list shows the root directory and the subdirectories on the drive you selected from the **Drive** list. By double-clicking on a subdirectory name, you can move to subdirectories further down the path. The files in the **File** list correspond to the current directory position in the list.

Required?: Yes.

Default: The current directory.

File

This list contains the files that you can select to be populated.

Your first or only selection appears in the **File name** field.

Required?: Yes. At least one selection is necessary.

Select All

Click this button to select all the files in the **File** list.

Deselect All

Click this button to deselect all the files in the **File** list.

IMS - Populate Notebook

This notebook enables you to:

⇑ Import information about one or more DBD or PSB source files into the TeamConnection database

⇑ Perform a trial run to preview the results of this process

The report displayed when an IMS DBD or PSB has been populated is stored in a file named DBD *nnnnn* .RPT or PSB *nnnnn* .RPT.   The file is stored in the directory specified on the General page of the Profile notebook.

Pages

DBD-PSB
TeamConnection

Push Buttons

Populate
Trial run
Cancel
Help

Populate

Click this button to start the process of populating objects into the TeamConnection database.

Trial Run

Click this button to get a report showing which objects would have been populated if you had clicked **Populate** .

The trial-run process does not store objects into the TeamConnection database.

DBD-PSB Page

This page enables you to:

⇑ Qualify the names of objects to be populated

⇑ Indicate whether the objects will be reconciled with shareable data elements during the populate process

Lists

Files selected
Name

Fields

Variation
Revision
Use mapping table

Check Box

Reconcile

Push Button

Search

File Selected

This list shows the names of the files you have selected to populate. Each entry is paired with an entry in the **Name** list.

Name

This list shows the default access names that correspond to objects in the **Files selected** list. The names of related objects that are also populated are not shown.

To change a default access name:

1. Hold down the ALT key and click the default name.
2. Type in the new file name.
3. Click anywhere in the **Name** list.

Variation

This field contains the name of a variation.

A variation is a qualifier that becomes part of the full name of an object in the TeamConnection database. This qualifier can be useful in distinguishing between similar objects that have different uses.

If an object is related to a higher-level object, it takes the variation name, if any, of the higher-level object. So, for example, a relational database object will automatically be assigned the variation name given to the higher-level system object when it was defined. Further, the same variation name will be assigned to table spaces within that database and to tables within one of the table spaces. Variation names are propagated in the same way from IMS DBDs to their segments and fields.

Required?: No.

Default: Initialized from the profile.

Revision

This field contains the name of a revision.

A revision is a qualifier that becomes part of the full name of an object in the TeamConnection database. This qualifier can be useful in distinguishing between similar objects that have to be available at the same time.

If an object is related to a higher-level object, it takes the revision name, if any, of the higher-level object. So, for example, a relational database object will automatically be assigned the revision name given to the higher-level system object when it was defined. Further, the same revision name will be assigned to table spaces within that database and to tables within one of the table spaces. Revision names are propagated in the same way from IMS DBDs to their segments and fields.

Required?: No.

Default: Initialized from the profile.

Reconcile

Check this box if you want DataAtlas to define populated segments and fields in terms of shareable data components identified in a mapping table. DataAtlas will create shareable data components if they don't already exist in the TeamConnection database.

TeamConnection Page

This page enables you to enter TeamConnection identifiers for the objects you're populating.

Fields

- Family
- Release
- Work area
- Component

Push Button

- Search

Family

This field contains the name of a family.

A family is a complete and self-contained collection of TeamConnection users and development data. Data within a family is isolated from data in other families.

Required?: Yes.

Restriction: The family must exist.

Default: Initialized from the profile.

Release

This field contains the name of a release.

A release is a logical organization of all the parts related to an application; that is, all the parts that must be built, tested, and distributed together.

Required?: Yes.

Restriction: The release must exist within the family specified.

Default: Initialized from the profile.

Limit: 15 characters.

Work Area

This field contains the name of a work area.

A work area is a view of a TeamConnection release. Release parts can be checked out to a work area, updated, and used in builds without affecting the integrated version of the parts in the release.

Required?: Yes.

Restriction: The work area must exist within the release specified.

Default: Initialized from the profile.

Limit: 15 characters.

Component

This field contains the name of a component.

A component is a logical group of development data within a TeamConnection family. Identifying a component is a means of controlling access to development data.

Required?: Yes.

Restriction: The component must exist.

Default: Initialized from the profile.

Limit: 31 characters.

Included Source Definition Notebook

This notebook enables you to create an included source definition, view a populated included source definition, and maintain both.   It contains general information about the object, its structure, and the definitions of objects in its structure.

Menu-Bar Choices

- IncludedSource
- Window
- Help

Pages

- General
- Tree View

Push Buttons

- OK
- Save
- Cancel
- Help

IncludedSource

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

General Page

This page enables you to name and describe in text the included source definition you are defining.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the included source definition.

Required?: Yes.

Variation

This field contains the name of a variation.

A variation name becomes part of the full name of an object in the TeamConnection database. It can be useful in distinguishing between similar objects that have different uses.

Required?: No.

Revision

This field contains the name of a revision.

A revision is a qualifier that becomes part of the full name of an object in the TeamConnection database.   This qualifier can be useful in distinguishing between similar objects that have to be available at the same time.

Required?: No.

Tree View Page

This page enables you to define or redefine the objects nested in an included source definition. It's divided into three parts: the **Representation** list, the *tree view* (left-hand side), and the details area (right-hand side), which changes according to the representation and the object you select from the tree view. You use the details area to define the selected object.

The tree view consists of a *root* (the included source name) and its *children* (objects nested within the root). A child can be:

⇑ A local data element or shareable data element. The latter can be either at the top level or at the 77 level of the included source. In either case, the child can be represented by an elementary or COBOL level-77 object.

⇑ A shareable data structure at the top level of the included source. It's represented by a group item.

⇑ A COBOL level-66 object, for renaming.

⇑ A COBOL level-88 object, for adding condition values to a parent object.

List

Representation

Tree View

Push Buttons:

Add sibling
Add child
Remove

Radio Buttons:

Elementary
Group
Level 66 (COBOL only)
Level 77 (COBOL only)
Level 88 (COBOL only)

Details Area

Search Box:

Use existing shareable data element/structure
Shareable data element/structure
Search
Open

COBOL details:

Name
RENAMES and THRU
REDEFINES
USAGE
PICTURE
SIGN
VALUE
GLOBAL
EXTERNAL
JUSTIFIED
SYNC
BLANK WHEN ZERO
OCCURS (check box)
Additional COBOL details when **OCCURS** *box is checked:*
OCCURS and To (spin buttons)
DEPENDING ON
Ascending, Descending
Available data items
Key data items
INDEXED BY

PL/I details:

Name
DEFINED
POSITION
Type
Precision
Scale
Length
PICTURE
INITIAL
ALIGNED
EXTERNAL
STATIC, CONNECTED, AUTOMATIC, and BASED
Arrayed
Additional PL/I details when **ARRAYED** *box is checked:*
Array dimensions
Add
Edit
Remove

Representation

Select either **COBOL** or **PL/I** . The page will be configured to let you define this object in the terms of the language you choose.

Default: Whichever representation was used last. On the first use, the default is taken from the profile.

Add Sibling

Click this button after you select an object in the tree, and DataAtlas adds another object below it.   Which type it adds depends on the radio button that is selected at the time.

If the root is selected, this push button is disabled.

Add Child

Click this button to add an object subordinate to the selected object. If the selected object is the root, you can add an elemental, group, or level-77 object as a child. Elemental and group objects can have only level-66 and level-88 objects added as children. Level 66s, 77s, and 88s cannot have children.

Remove

Click this button after you select a child object in the tree, and DataAtlas removes it from the tree.

If the root is selected, this push button is disabled.

Elementary

Select this button to add an elementary object to the tree, either as a child of the root or as a sibling of a selected child.

An elementary object can be defined by a shareable data element.

Group

Select this button to add a group object to the tree, either as a child of the root or as a sibling of a selected child.

A group object in an included source **must** be defined by a shareable data structure. If you populate a COBOL COPY file or PL/I include file that contains a group object, DataAtlas creates a shareable data structure that defines the group object.

Level 66 (COBOL only)

Select this button to add a level-66 object as a child of an SDS.   You'll then have to add RENAMES information about objects nested within the included source definition, the SDS, or both.

Level 77 (COBOL only)

Select this button to add a level-77 object to the tree, either as a child of the root or as a sibling of a selected child.

A level-77 object can be defined by a shareable data element.

Level 88 (COBOL only)

Select this button to add a level-88 object to the tree, either as a child of an elemental or group object or as a sibling of another level-88 object.

A level-88 object adds information about conditional values to its parent object.

Use Existing Shareable Data Element/Structure

If you selected an elementary object in the tree view, you can check this box to use a shareable data element to define the elementary object. When you check the box, the **Search** push button is enabled.

If you selected a group object in the tree view, no check box appears. This indicates that a group object **must** be defined by a shareable data structure.

Default: Unchecked.

Shareable Data Element/Structure

If you search for and select a shareable data component, DataAtlas puts the name of the selected component in this field.

Search

Click this button to see a list of shareable data components (whether elements or structures depends on the selected object). If you select a shareable data component from the list, it is used to define the selected local data component in the tree. The name of the shareable data component will appear in the **Name** field inside the search box and will overwrite a FILLER name in the **Name** field outside the search box. If the name outside the search box is changed, the label of the selected object in the tree will change accordingly.

Open

Click this button to open the notebook for the shareable data element or structure identified in the adjacent **Name** field.

Name

This field contains the name of the selected object in the tree view. If you are creating an included source definition, enter the name of each object here. Afterward, or after an included source definition has been populated, you can rename a selected object by editing this field and saving the definition.

If you use a shareable data component to define an object in the tree, DataAtlas puts the name of the component in this field. Editing the name doesn't rename the shareable data component; the new name becomes an alias for the component.

Required?: No.

Default: 'FILLER' for COBOL, '*' for PL/I.

Note: If this field is empty when a COBOL included source definition is stored, the name is stored as 'FILLER'.

RENAMES and THRU

Select a **RENAMES** item from the list to indicate the first item being renamed. Select a **THRU** item to indicate the last item being renamed. These items and all those between them will be renamed with the name in the **Name** field. If you leave the **THRU** field blank, only the item in the **RENAMES** field will be renamed.

These two fields appear only for COBOL level-66 objects.

Required?: **RENAMES** is; **THRU** is not.

REDEFINES

Check this box if you want to redefine the previous object in the tree in terms of the current object.

When you check the box, the name of the object that can be redefined is filled in. If the REDEFINES attribute was set in the PL/I representation, the field may contain a different object name that may not be valid in COBOL terms. To reset the value to the valid COBOL REDEFINES object, uncheck and recheck the box.

Note: You can't check this box for the first object in the tree in a COBOL representation, because it has no preceding object.

USAGE

If you are defining an object in the tree, select a USAGE clause value from this list.

Required?: Yes.

Default: DISPLAY.

PICTURE

If you are defining an object in the tree, enter a PICTURE clause value in this field.

Required?: Yes, for BINARY, DISPLAY, DISPLAY-1, and PACKED- DECIMAL usage clauses.

Default: Depends on the USAGE clause.

SIGN

If you are defining an object in the tree, you can select a SIGN clause value from the list only for a **USAGE** value of DISPLAY and a value for a signed numeric number type in the **PICTURE** field.

Required?: No.

VALUE

This field can specify the initial contents of the selected object. It serves the same purpose as the INITIAL field in a PL/I representation. However, a value in either field may not be valid in the other representation. This is especially true for string, array, and pointer values.

If the selected object is a level-88 object, this field can specify a single value, multiple values, a range of values, multiple ranges of values, or a combination of ranges and values. The contents of this field are assigned to the condition name in the **Name** field.

Enclose values for strings in double quotes. Enter only one value for an array; the value is used for each arrayed element. Enter NULL or NULLS for a pointer.

DataAtlas doesn't check the syntax of this field when you enter a value, but it does validate the contents when you generate a COPY file.

Required?: Only for level-88 objects.

GLOBAL

Check this box if the selected object is available to every program contained in the program that declares it.

Default: Unchecked.

EXTERNAL

Check this box if the storage associated with the selected object is associated with the run unit rather than with any particular program or method within the run unit.

Default: Unchecked.

JUSTIFIED

Check this box if the selected object is defined for alphabetic or alphanumeric data, and you want the data to be aligned at the rightmost character position.

Default: Unchecked.

SYNC

Check this box if the selected object should be aligned on a natural boundary in storage.

Required?: No.

BLANK WHEN ZERO

Check this box if the selected object is an elementary object and you want it to contain nothing but spaces when its value is zero.

Required?: No.

OCCURS (Check Box)

Check this box if the selected object in the tree is a table element that can be referred to by indexing or subscripting.

Note: Generating a COBOL included source definition with OCCURS on a level-1 object produces invalid COBOL source. Therefore, when you generate an included source definition that has an OCCURS clause on one of its children, the start level for the generation should be greater than 1.

Type

Select a data type from this list. The **SIGNED** and **UNSIGNED** radio buttons apply only to the FIXED BIN data type. If **SIGNED** is selected, the fixed binary variable can have a negative sign. The **REAL** and **COMPLEX** radio buttons apply only to the FIXED BIN, FIXED DEC, FLOAT BIN, FLOAT DEC, and PICTURE data types.

Required?: Yes.

Default: CHARACTER.

Precision

Use the adjacent spin button to specify the number of digits that FIXED BIN, FIXED DEC, FLOAT BIN, or FLOAT DEC data can have.

Scale

This field applies only to the FIXED BIN and FIXED DEC data types. Use the adjacent spin button to specify the position of the decimal or binary point relative to the rightmost digit.

Length

This field specifies the length in bytes of CHARACTER, BIT, or GRAPHIC data. The default is 1 byte.

These radio buttons give further information about length:

⇑ **VARYING** says that the length of the data can vary; **NONVARYING** says it can't.

⇑ **VARYINGZ** says that the length of CHARACTER or GRAPHIC data can vary, with a zero byte at the end.

PICTURE

This field contains a picture specification when the data type is PICTURE.

INITIAL

This field contains the initial value or values of a variable. It serves the same purpose as the VALUE field in a COBOL representation . However, a value in either field may not be valid in the other representation. This is especially true for string, array, and pointer values.

Specify only one value for an elementary object, unless it's an array variable; then you can specify multiple values. Separate values for arrayed elements with a comma.

Enclose values for strings in single quotes. Enter a varying flag **and** a string value if you are initializing a pointer. Examples:

```
    VAR)('aString'
    NONVARYING)('aStringX'
```

Notice the absence of opening and closing parentheses. These are added when you generate the include file.
DataAtlas doesn't check the syntax of this field when you enter a value, but it does validate the contents when you generate the include file.

Limitation: **INITIAL** values are not carried forward when you populate a PL/I include file. However, the value you specify here will be used when you generate an include file.

EXTERNAL

Check this box it the name of the selected element or structure can be known outside the declaring block. An unchecked box indicates that the element or structure is internal.

Default: Unchecked.

STATIC, CONNECTED, AUTOMATIC, and BASED

Select the button that defines the storage class of the element or structure. **AUTOMATIC** and **BASED** can be selected only if the **EXTERNAL** box is unchecked.

If you select **BASED** , you can enter the name of an object on which the selected object is based. You can explicitly name an object already in the included source definition, or you can name a pointer object that will then be implicitly or contextually defined in the included source. You can also choose to leave this field empty.

Limitation: **BASED** specifications are not carried forward when you populate a PL/I include file. However, what you specify here will be used when you generate an include file.

Information Report Window

This report shows the results of the performed actions for getting information on potential design problems for the selected objects.

The results are presented in terms of information report items, which consist of an icon and a detailed description.

Each information report item informs you on a design step to consider during the database design of the selected objects.

On request, you get more information on an information report item.

Also, you can look at the rule which caused a report item.

Tasks

Evaluating the design support report

Push Buttons

Cancel
Help

Menu-Bar Choices

Report
Window
Help

Pop-up menu

Rule
More

Report

This choice displays one action you can apply to the displayed report:

Save as...	Allows you to specify a name for the report, and where you want to store it.

Rule

Select this choice of the pop-up menu to get the detailed description of the rule on which the action item is based.

More

Select this choice of the pop-up menu to look at the objects affected by the performance of the action item.

Information Report Item Window

A report item reports the result of the performance of a previously selected action item.

You can request this window during the evaluation of a design support report.

In this window, you see the detailed description of an information report item, and the objects affected by the performance of the action item.

Tasks

You can look at the notebook of an affected object by double-clicking on its icon.

Push Buttons

- Cancel
- Help

Mapping Table - Search Window

This window enables you to specify the file containing the mapping table to be used in a reconcile operation.

Field

Open filename

Lists

Type of file
Drive
Directory
File

Push Buttons

OK
Cancel
Help

Open Filename

This field identifies the file that contains the mapping table you want to use. You can fill in the field by typing into it directly or by clicking an entry in the **File** list.

When you click **OK**, the name in this field is entered into the **Use mapping table** field in the notebook you came from.

Required?: Yes.

File

Click a file name in this list and it's entered into the **Open filename** field. If you double-click a file name, it's entered directly into the **Use mapping table** field in the notebook you came from; this window closes.

Move Objects Window

Select the workfolder into which you want the objects moved. When you select the **OK** push button, the objects are transferred from the source workfolder to the target workfolder.

If the target workfolder is open, the icons for the objects appear in it. When you close the workfolder, the changes you made are stored. If the target workfolder is closed, DataAtlas moves the objects into it and stores the changes.

Restriction: The target workfolder must belong to the same family, release, and work area as the source workfolder.

List

- Objects

Push Buttons

- OK
- Cancel
- Help

Objects

This read-only list identifies the objects that will be moved.

Object List Window

This window shows the database objects contained in the selected object class.

You can either directly design an object by opening the corresponding notebook into which you enter the object's database design data. You can only open the notebook of one object at a time.

Or, you can select one or more objects for which you want to get design support.

Tasks

- Opening the notebook of a database object for direct design and maintenance
- Requesting design support for one or more objects

Menu-Bar Choices

Designer
Edit
View
Window
Help

Designer

This menu-bar choice displays a pull-down menu with the types of design support for the object you are designing:

Inform	Gives you information on potential design problems.
Propose	Offers proposals for selected design items and executes them on request.
Validate	Validates your current design.

The pull-down contains only those items that apply to the object you are designing. For example, for some objects you only get information on potential design problems.

View

This choice displays two actions you can apply:

Icons	Displays all of the items as icons.
Details	Displays all of the items in a details view.

Open Window

This window enables you to start an editing session with another file.

Field

Open filename

Lists

Type of file
Drive
Directory
File

Push Buttons

OK
Cancel
Help

Open Filename

This field identifies a file you want to open and edit. Fill it in by overtyping the name of the report file or by clicking a name in the **File** list.

Required?: Yes.

Drive

Select from the drop-down list the drive where the file resides or will reside.

Required?: Yes.

Default: The drive of the current directory.

File

Click a file name in this list and it's entered into the **Open filename** field.

Oracle Column Notebook

This notebook enables you to define the data attributes of a new column or modify the attributes of an existing column.

Page

Definition

Push Buttons

OK
Save
Cancel
Help

Definition Page

This page enables you to define and update the information for a column. It also lets you connect a shareable data element to the column.

Fields

- Name
- Comment
- Shareable Data Element Name
- Length
- Description

Lists

- Type
- Precision
- Scale

Push Buttons

- Search
- Open
- Remove

Check Box

- Null not allowed

Name

This field identifies the name of a column.

Required?: Yes.

Limit: 18 characters.

The column name must be in a valid format.

Length

This field contains the maximum length of the column's data.

Default: 1.

Null Not Allowed

Check this box if the column cannot contain nulls.

Oracle Index Notebook

This notebook enables you to create or modify an index for an Oracle table.

Pages

- Definition
- General
- Option

Push Buttons

- OK
- Save
- Cancel
- Help

Definition Page

This pages enables you to:

⇑ Search the TeamConnection database for a table whose index you want to create or modify

⇑ Specify which columns you want to be part of the index

⇑ Prevent index columns from having duplicate values

Table

Table Name
Search
Open

Lists

Available columns
Index columns

Push Buttons

Add
Remove

Radio Buttons

Unique Constraint

General Page

This page enables you to enter general information about an Oracle index. The **System** and **Schema** fields cannot be modified for an existing index.

Fields

- System
- Schema
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Schema

This field contains the user ID under which the index was created. When a schema is selected by using the **Search** push button, both this field and the **System** field are initialized.

Required?: Yes.

Name

This field contains the access name of the Oracle index.

Required?: Yes.

Option Page

On this page you can connect the index to or disconnect it from an existing table space and physical design.

Field

- Table space

List

- Physical design

Push Buttons

- Search
- Open
- Remove

Table Space

If you want to connect the index to a table space, select the adjacent **Search** push button and a table space from the resulting list. This field is filled in with the name of the table space you choose. Then open the table space (**Open** push button) and identify this index in the **Table Space** notebook.

Required?: No.

Physical Design

This is a list of physical designs to which the index belongs. You can connect the index to a new physical design by selecting the adjacent **Search** push button and a physical design from the resulting list. The name of the selected physical design will be added to the list on this page. Then open the selected physical design (**Open** push button) and identify this index in the **Physical Design** notebook.

Required?: No.

Search

Click this button to see a list of table spaces or physical designs to which you can connect the index. The list of table spaces contains only those that belong to the relational system this index belongs to. (The name of the relational system is in the **System** field on the **General** page.)

Oracle Physical Design Notebook

This notebook enables you to open the properties notebooks of the objects that belong to this Oracle physical design.

The notebook' pages correspond to the objects that can be in an Oracle physical design - tables, table spaces, indexes, and views.

Menu-Bar Choices

Physical design
Window
Help

Pages

General
Index
Table
Table Space
View

Push Buttons

OK
Save
Cancel
Help

General Page

Use this page to specify general information about an Oracle physical design.

Fields

- Relational design
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Index Page

This page enables you to view the list of indexes that belong to the physical design and open the properties notebook for an entry selected from the list.

List

- Indexes

Push Button

- Open
- Search
- Remove

Table Page

This page enables you to view the list of tables that belong to the physical design and open the properties notebook for any entry selected from the list.

List

- Tables

Push Button

- Open
- Search
- Remove

Table Space Page

This page enables you to view the list of table spaces that belong to the physical design and open the properties notebook for any entry selected from the list.

List

- Table spaces

Push Button

- Open
- Search
- Remove

View Page

This page displays the list of views that belong to the physical design. You can open the properties notebook for an entry selected from the list.

List

- Views

Push Button

- Open
- Search
- Remove

Oracle Table Notebook

This notebook enables you to define and maintain an Oracle table in the TeamConnection database.

Pages

- Definition
- Primary Key
- Unique Keys
- Foreign Keys
- Checks
- Options
- Indexes
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Primary Key Page

On this page you can give the primary key a name and choose which column or columns define the primary key for the table. The columns you choose must not be the same as those in a unique key.

Field

- Constraint name

Lists

- Available columns
- Primary key columns

Push Buttons

- Add
- Remove

Available Columns

This list contains all the columns that qualify to be used in a primary key. The columns in the list are columns that are not used in the primary key already.   They   must be defined as NOT NULL or NOT NULL WITH DEFAULT and cannot have a data type of LONG or LONG RAW.

Unique Keys Page

On this page you can identify the unique keys for the table.

Lists

- List of unique keys
- Available columns
- Unique key columns

Field

- Constraint name

Push Buttons

- Add
- Remove
- Create
- Modify
- Delete

List of Unique Keys

This list identifies all the unique keys for the table. Select the entry &lt.NEW> if you want to identify a new unique key. The fields below the list will be cleared for your use. Click **Create** after you select and name the unique key.

Select an existing unique key if you want to update the fields below this list. Then click **Modify** to apply your update.

Restriction: You can't designate the same column or columns as both a unique key and a primary key.

Available Columns

This list identifies all the columns of the table. To choose a unique key, select one or more columns that do not have a data type of LONG or LONG RAW. Then click **Add** . The selected columns will be moved to the **Unique key columns** list.

Unique Key Columns

This list contains the column names you select from the **Available Columns** list.

Limit: A composite unique key can contain a maximum of 16 columns.

Constraint Name

This field contains the name of the unique key.

Required?: Yes.

Default: If you don't specify a name, DataAtlas will generate a nondescriptive name from the series UniqueKey1, UniqueKey2, ...

Limit: 30 characters.

Add

Click this button to move selected columns from the **Available columns** list to the **Unique key columns** list.

Remove

Click this button to remove selected columns from the **Unique key columns** list and return them to the **Available columns** list.

Create

Click this button to identify a new unique key after you have:

1. Clicked the entry marked &lt.NEW> in the list of unique keys

2. Selected the column or columns for the unique key

3. Optionally named and written a comment about the unique key

Modify

Click this button to modify a unique key after you have:

1. Clicked the entry in the list of unique keys

2. Updated the information that defines the unique key

Delete

Select an entry in the **List of unique keys** and click this button to delete the entry.

Foreign Keys Page

This page enables you to add and remove a foreign key. To add a foreign key, click &lt.NEW> in the foreign key list, define the new foreign key, and click **Create** . To remove a foreign key, select an existing foreign key and click **Delete** .

Lists

- List of foreign keys
- Primary key
- Matching columns

Fields

- Constraint name
- Referenced table

Check Box

- Delete cascade

Push Buttons

- Create
- Delete
- Search

Delete Cascade

Check this box if you want the database to maintain referential integrity by automatically removing a foreign key value when a referenced primary or unique key value is removed.

Default: Unchecked.

Checks Page

This page enables you to define and maintain the check constraints for one or more columns of a table.

List

- List of check constraints

Fields

- Constraint name
- Check condition

Push Buttons

- Create
- Modify
- Delete

Options Page

On this page you can connect the table to or disconnect it from an existing table space and physical design.

Field

- Table space

List

- Physical design

Push Buttons

- Search
- Open
- Remove

General Page

This page enables you to enter and update general information about an Oracle table. The **System** and **Schema** fields cannot be modified for an existing table.

Fields

- System
- Schema
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

System

This read-only field contains the name of the Oracle SQL server.

Schema

This field contains the user ID under which an object in the relational system was created. When a schema is selected by using the **Search** push button, both this field and the **System** field are initialized.

Required?: Yes.

Name

This field contains the access name of the Oracle table.

Required?: Yes.

Oracle Table Space Notebook

This notebook enables you to view the tables and indexes in this table space.

Pages

- Contents
- General
- Option

Push Buttons

- OK
- Save
- Cancel
- Help

Contents Page

This pages enables you to select a table or index and open its settings notebook.

List

Usage/Name

Push Button

Open

Usage/Name

This list identifies the tables and indexes in this table space. To view the properties notebook for a listed table or index, select it, and then select the **Open** push button.

If a table is connected to the table space, the Usage column indicates a **Primary** connection. If a table's indexes are connected to the table space, it indicates an **Index** connection. If a table's long data is connected to the table space, it indicates a **Long** connection.

Open

Click this button to view the properties notebook of a table or index that you've selected from the **Usage/Name** list.

General Page

This page enables you to enter general information about an Oracle table space.

Fields

- System
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the Oracle table space.

Required?: Yes.

Option Page

On this page you can connect the table space to or disconnect it from an existing physical design.

List

Physical designs

Push Buttons

Search
Open
Remove

Oracle View Notebook

This notebook enables you to display and modify a view's SQL definition and columns, and to comment on the view.

Pages

- Definition
- General
- Option

Push Buttons

- OK
- Save
- Cancel
- Help

Definition Page

This pages enables you to:

⇑ See the SQL definition of a view
⇑ Add, modify, or delete view columns
⇑ Add a comment for a view and its columns

Fields

- Fullselect
- Comment (on the SQL statement)
- Name
- Comment (on a view column)

List

- Columns

Push Buttons

- Add
- Modify
- Delete

General Page

This page enables you to enter general information about an Oracle view. The **System** and **Schema** fields cannot be modified for an existing view.

Fields

- System
- Schema
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Option Page

On this page you can:

⇑ Define characteristics of the view

⇑ Connect the view to or disconnect it from an existing physical design

Check Boxes

- FORCE
- WITH

List

- Physical designs

Push Buttons

- Search
- Open
- Remove

FORCE

Check this box if you want the view to be created whether or not the view's base tables exist or the owner of the schema containing the view has privileges on the tables.

Required?: No.

WITH

Check this box to enable the **READ ONLY** and **CHECK OPTION** radio buttons. **READ ONLY**, the default, specifies that no deletes, inserts, or updates can be performed through the view. **CHECK OPTION** specifies that inserts and updates performed through the view must result in rows that the view query can select. In the adjacent **Constraint name** field, you must enter a name for the **CHECK OPTION** constraint.

Required?: No.

Physical Designs

This is a list of physical designs to which this view belongs. You can connect the view to a new physical design by selecting the adjacent **Search** push button and a physical design from the resulting list. The name of the selected physical design will be added to the list on this page.

Required?: No.

Search

Click this button to find a physical design to which you can connect this view.

Open

Click this button to open a **Physical Design** notebook for a selected physical design.

Remove

Click this button to remove the selected physical design from the list and disconnect this view from it.

PCB Notebook (All PCBs)

This notebook enables you to create and update the definition of an IMS process control block (PCB).

PCBs can be shared by many PSBs.

Menu-Bar Choices

- PCB
- Window
- Help

Pages

- DB
- GSAM
- TP
- General

Push Buttons

- OK
- Save
- Cancel
- Help

PCB

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

DB Page

This page enables you to define a database PCB.

Fields

- DBD Name
- Processing Options
- Key Length

List

- Sequence DBD

Check Box

- List

Radio Buttons

- Positioning
- Sequential Buffering

Table

- Sensitive Segments

Push Buttons

- Search
- Create
- Open
- Delete

Base DBD

This field contains the name of the base DBD.

Required?: Yes.

Processing Options

This field specifies the processing options to be used with the segments identified in the **Sensitive Segments** table. Each of the following letters represents a processing option. Enter your choices as a consecutive string of letters; for example, **GON** .

G. Get function.

I. Insert function.

R. Replace function. (includes **G.**)

D. Delete function. (Includes **G.**)

A. All of the foregoing functions.

P. Required if command code D is used, except for ISRT calls in a batch program that is not sensitive to fields. (For Fast Path applications, **P** is not required if command code D is used when processing DEDBs.)

O. Read only. IMS is not to check the ownership of the segments returned. Therefore, a read-without-integrity program might get a segment that has been updated by another program.

N. Prevents a read-only application program from abending if it detects an invalid pointer.

T. The same as **N** , except that this causes DL/I to automatically retry the operation.

E. Gives exclusive use of the database or segment to online programs.

L. Local function for database loading, except HIDAM.

GS. Get segments in ascending sequence only. (Applies only to HSAM.)

LS. Load segment in ascending sequence only. (Applies only to HIDAM and HDAM; required for HIDAM.)

H. Specifies high-speed sequential processing. (Applies only to BMPs that process DEDBs.)

Default: **A** .

Key Length

This field specifies the length in bytes of the longest concatenated key for a hierarchic path of sensitive segments that the application program uses in a logical data structure.

Required?: Yes.

Sequence DBD

If this list is initialized, secondary indexes are used to process the database named in the **Base DBD** field. Select a secondary index from the list.

Required?: No.

List

Check this box if you want to include this PCB in a PCB list that is passed to the application program at entry. Otherwise, do not check it.

Default: Checked.

Positioning

Select the **Single** button to specify single positioning for the logical data structure; select the **Multiple** button for multiple positioning.

Default: **Single**, except for DEDBs with more than two dependent segments. Then it is **Multiple**.

Restriction: HSAM does not support multiple positioning.

Sequential Buffering

Select the **Conditional** button for conditional sequential buffering. This choice lets IMS activate sequential buffering when it detects a sequential I/O reference pattern and a reasonable activity rate.

Select the **No** button if you want no sequential buffering.

Default: **No** .

Recommendation: For short-running MPPs, Fast Path programs, and CICS programs, select **No** .

Sensitive Segments

This table lists the segments to which a program is sensitive through this PCB.

You can use the **Create** push button add a new sensitive segment to this table.

The **Open** push button lets you open the **Sensitive Segment** notebook for the entry you highlighted in the table. You can then review the segment's definition or redefine it.

The **Delete** push button lets you delete from the table the entries you highlighted.

Create

Click this button to open a **Sensitive Segment** notebook and define a new sensitive segment for this PCB.

Open

Click this button to open the **Sensitive Segment** notebook for the entry you highlighted in the table. You can then review the segment's definition or redefine it.

Delete

Click this button to delete from the **Sensitive Segments** table the entries you highlighted.

GSAM Page

This page enables you to define a GSAM PCB.

Fields

- Base DBD
- Processing Options

Check Box

- List

Push Button

- Search

Processing Options

This field specifies the processing options that can be used by an application program on the database identified in the **Base DBD** field. Each of the following letters represents a processing option.

G. Get function.

L. Load function.

S. Large-scale sequential activity. (Use the GSAM multiple-buffering option.)

Required: Yes.

TP Page

This page enables you to define an alternate (teleprocessing) PCB.

Fields

- Output Destination
- Output Type

Check Boxes

- Modify
- List
- Alternate response
- Same terminal
- Express

Output Destination

This field specifies the destination of output messages. Therefore, it contains either the name of a logical terminal or a transaction-code name. Which one you enter is determined by your choice in the **Output Type** field.

Required?: Yes, unless the **MODIFY** box is checked.

Output Type

For this field, select **LTERM** if the destination of output messages is a logical terminal. Select **NAME** if the destination is a transaction-code name.

Required?: Yes, unless the **MODIFY** box is checked.

Modify

Check this box if you want this PCB to be modifiable.   This means the output destination name associated with this PCB can be dynamically modified.

Default: Unchecked.

Alternate Response

Check this box to enable this PCB to be used instead of the I/O PCB for responding to terminals in response mode, conversational mode, or exclusive mode.

Default: Unchecked.

Same Terminal

Check this box if you want IMS to verify that the logical terminal named in the response alternate PCB is assigned to the same physical terminal as the logical terminal that originated the input message.

Default: Unchecked.

Express

Check this box if you want messages from this PCB to be sent if the application program abends. Do not check it if you want the messages to be backed out instead.

Default: Unchecked.

General Page

This page enable you to name the PCB and write a description of it.

Fields

- Data type
- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Set Type
- Search

Data Type

This field identifies the type of PCB this is.

For a new PCB, select a type from the list below this field. When you select the **Set Type** push button, the type you have chosen is locked in.

Name

This field contains the access name of the PCB.

Required?: Yes.

Limit: 8 alphanumeric characters.

Set Type

Click this button to lock in the type you have chosen for this PCB.

PL/I Array Dimensions Window

Use this window to specify a new dimension or respecify an existing dimension in the Array dimensions table of the **Included Source Definition** or **Shareable Data Structure** properties notebooks.

Fields

- DIMENSION
- Lower bound
- Upper bound

Push Buttons

- OK
- Cancel
- Help

DIMENSION

If you came to this window by selecting **Edit** in the properties notebook, set the spin button to the dimension number you want to edit. By default, it is set to the currently selected dimension number.

If you came here by selecting **Add** , set the spin button to the dimension number that precedes the number you want to add. By default, it is set to 1 more than the highest dimension number.

When a new dimension is inserted, the dimension numbers that follow it are incremented.

Lower Bound

Specify the dimensions' lower bound in this field.

Required?: No.

Default: 1.

Upper Bound

Specify the dimensions' upper bound in this field.

Required?: Yes.

Default: 1.

PL/I - Populate Notebook

This notebook enables you to:

⇑ Import information about one or more PL/I include files into the TeamConnection database

⇑ Perform a trial run to preview the results of this process

The report displayed when a PL/I include file has been populated is stored in a file named PLI *nnnnn* .RPT. The file is stored in the directory specified on the General page of the Profile notebook.

Pages

PL/I
Compiler
TeamConnection

Push Buttons

Populate
Trial run
Cancel
Help

PL/I Page

This page enables you to:

⇑ Qualify the names of objects to be populated

⇑ Indicate whether the objects will be reconciled with shareable data components during the populate process

Lists

Files selected
Name

Fields

Variation
Revision
Use mapping table

Check Box

Reconcile

Push Button

Search

Compiler Page

This page enables you to specify compiler options by entering them directly or by referring to a command file that contains them.

Fields

Compile options
Command file

Check Box

Use command file not compiler options

Push Button

Search

Compile Options

This field contains the compiler options you want to use in the compilation phase of populating a PL/I include file. Alternatively, you can put compiler options in a command file.

Required?: No.

Default: Initialized from the profile.

Restriction: If you specify additional compiler options, they must not conflict with the default set.

Command File

This field identifies a file that invokes the compiler and passes it compiler options when a PL/I include file is populated. Alternatively, you can specify compiler options in the **Compile options** field.

The **Use command file not compiler options** box enables the **Command file** field.

Required?: No.

Default: Initialized from the profile.

Note: The path `&lt.DataAtlasInstallPath>\lang\en_us\samples\pli` contains a sample command file named `ewslppg.cmd` . `&lt.DataAtlasInstallPath>` is the root directory where DataAtlas is installed.

Populate from... Folder

From this folder you can access your source files and select the ones you want to populate into the TeamConnection database.

Menu-Bar_Choices

- Folder
- Selected
- Edit
- View
- Help

Icons

- IMS Source
- COBOL Source
- PL/I Source
- DB2 Source

Profile Folder

A profile lets you specify default settings for your work environment and your use of DataAtlas.   This folder is the starting point for creating, modifying, and deleting profiles.

Menu-Bar Choices

Folder
Selected
Edit
View
Window
Help

Icon

DA.INI (Active)

View

This choice controls the appearance of the icons in the folder. The actions on its menu are:

Icon view. Shows objects as icons.

Flowed icon view. Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Sort. Offers the choice **Name,** which reorders the icons alphabetically by name.

Arrange. Arranges the icons to fit the size of a resized window.

DA.INI (Active)

You can use this icon in two ways:

Double-click it. This opens the profile that comes with DataAtlas. The profile has the form of a notebook; it resides in the file DA.INI.

Click it with the right mouse button. This opens a pop-up menu with the options **Settings** and **Help**. **Settings** opens the profile (equivalent to double-clicking the icon). **Help** shows you the same pull-down menu that is under the **Help** menu-bar choice.

Profile Notebook

This notebook enables you to view and change the profile options that are used throughout DataAtlas. Its pages are divided according to the data management technologies that DataAtlas supports.

DA.INI is the file name of the default profile. If it doesn't exist (the case when you invoke DataAtlas for the first time), DataAtlas automatically creates one for you and prompts you to specify a TeamConnection family, release, work area, and component. You can modify these specifications by updating the **TeamConnection** page of this notebook.

Pages

- General
- TeamConnection
- Query
- DB2/390
- DB2/390 Designer
- DB2/390 Generate
- DB2 UDB
- DB2 UDB Generate
- Oracle
- Oracle Generate
- IMS
- IMS Generate
- COBOL
- COBOL Generate
- PL/I
- PL/I Generate
- Physical Design

Push Buttons

- OK
- Cancel
- Save
- Help

COBOL Page

This page enables you to set the defaults that DataAtlas uses when COBOL Copy files are populated.

Fields

- Compiler options
- Command file
- SPECIAL NAMES
- Source directory
- Prefix
- Suffix
- Use mapping table

Check Boxes

- Use command file not compiler options
- Reconcile

Push Buttons

- Search
- Reset

Compiler Options

This field contains default compiler options to be used in the compilation phase of populating a COBOL COPY file.   Alternatively, you can put compiler options in a command file.

Required?: No.

Restriction: If you specify additional compiler options, they must not conflict with the default set. For example, you cannot specify NOADATA as a compiler option.

Command File

This field identifies a file that contains the default compiler options to be used in the compilation phase of populating a COBOL COPY file. Alternatively, you can specify compiler options in the **Compile options** field.

The **Use options in command file** box enables the **Command file** field.

Required?: No.

SPECIAL NAMES

Enter in this list any of the clauses of the SPECIAL NAMES paragraph that you want to give to the compiler.

Required?: No.

Source Directory

This field identifies a directory that contains COBOL COPY files to be populated.

Required?: No.

Default: The current directory.

Prefix

This field can contain a default string to be stripped from the beginning of a COBOL data name during population. If you enter lowercase alphabetical characters, they will be converted to uppercase before they are used.

Required?: No.

Suffix

This field can contain a default string to be stripped from the end of a COBOL data name during population. If you enter lowercase alphabetical characters, they will be converted to uppercase before they are used.

Required?: No.

Use Command File Not Compiler Options

Check this box if you want DataAtlas to ignore the **Compile options** field and use instead the options in a command file.

When you check this box, the **Command file** field and the **Search** push button are enabled.

Default: Unchecked.

Reconcile

Check this box if you want DataAtlas to define populated local data components in terms of shareable data components identified in a mapping table. DataAtlas will create shareable data components that don't already exist in the TeamConnection database.

COBOL Generate Page

This pages enables you to set defaults for generating COBOL COPY files.

Fields

- Output directory
- Prefix
- Suffix
- Start level
- Increment

Push Buttons

- Search
- Reset

Radio Buttons

- Never, On errors, Always

Output Directory

This field identifies the directory that will contain generated COBOL COPY files.   The drive must be included with the directory name.

Required?: No.

Default: The current directory.

Prefix

This field can contain a default string to be added to the beginning of a COBOL data name during file generation.

Required?: No.

Suffix

This field can contain a default string to be added to the end of a COBOL data name during file generation.

Required?: No.

Start Level

Use the spin buttons to select the starting level for the first element in generated files.

Default: 1.

Increment

Use the spin buttons to select the increment to be used for the second and succeeding elements in generated files.

Default: 5.

Never, On Errors, Always

Select the button that tells DataAtlas when to produce a report of the generate process - never, when an error occurs, or always.

Default: **Always** .

DB2/390 Page

This page enables you to set the defaults that DataAtlas uses when DB2/390 tables are populated.

Fields

- System
- Database
- Creator
- Relational design
- Physical design
- Use mapping table

Check Boxes

- Reconcile
- Use or create data elements for all columns

Push Buttons

- Search
- Clear
- Reset

System

This field contains the name of the system that corresponds to an instance of a DB2 catalog. For a distributed database, this is the location ID of the requesting DBMS or a DB2 subsystem. Otherwise, this is a name that uniquely identifies a relational catalog.

Required?: No.

Restriction: The system object must already exist in the TeamConnection database.

Database

This field contains the alias of the database containing the objects (tables, views, indexes) to be populated.

Required?: No.

Restriction: The database object must already exist in the TeamConnection database.

Creator

This field contains the user ID of the table's creator.

Required?: No.

Restriction: The creator object must already exist in the TeamConnection database.

Relational Design

This field contains the name of a relational design object. Populated objects will be associated with this relational design object.

Required?: No.

Restriction: The relational design object must already exist in the TeamConnection database.

Physical Design

This field contains the name of a physical design object. Populated objects will be associated with this physical design object.

Required?: No.

Restriction: The physical design object must already exist in the TeamConnection database.

Search

Click this button to see a list of items that can be entered into the corresponding field.   If you select an item from the list, it will appear in the field.

DB2/390 Designer Page

This page enables you to describe the MVS system environment and set options for DataAtlas Designer.

Fields

- Database name
- Storage group
- Buffer pools
- Index suffix
- Index prefix
- Table space prefix
- Foreign key prefix

Check Boxes

- Use Storage Management Subsystem (SMS)
- Close data sets
- Use MIXED DATA

Push Button

- Reset

Database Name

This field identifies the current database to which an object under design is assigned.

Previously entered names are kept in a drop-down list.

Limit: 18 alphanumeric characters.

Default: DSNDB04.

Storage Group

This field contains the name of a storage group, a set of volumes on DASDs that holds the data sets in which tables and indexes are stored.

Previously entered names are kept in a drop-down list.

Limit: 18 alphanumeric characters.

Default: SYSDEFLT.

Buffer Pools

This field specifies the size and use of a buffer pool. A buffer pool in the range of BP0 to BP49 is 4K bytes in size and can be used for table spaces or indexes. A buffer pool in the range of BP32K to BP32K9 is 32K bytes in size and is only for table spaces.

Default: BP0.

Index Suffix

This field contains a suffix that you want DataAtlas Designer to add to the index names it generates.

Previously entered suffixes are kept in a drop-down list.

Limit: 6 characters.

Default: IN.

Index Prefix

This field contains a prefix that you want DataAtlas Designer to add to the index names it generates.

Previously entered prefixes are kept in a drop-down list.

Limit: 5 characters.

Default: IN.

Table Space Prefix

This field contains a prefix that you want DataAtlas Designer to add to the table space names it generates.

Previously entered prefixes are kept in a drop-down list.

Limit: 5 characters.

Default: TS.

Foreign Key Prefix

This field contains a prefix that you want DataAtlas Designer to add to the foreign keys it generates.

Previously entered prefixes are kept in a drop-down list.

Limit: 5 characters.

Default: FK.

Use Storage Management Subsystem (SMS)

Check this box if you are using the Storage Management Subsystem (SMS).

Default: Unchecked.

Close Data Sets

Check this box to show that the data set containing the table space or index can be closed when the table space or index is not in use and limit of open data sets has been reached.

Default: Checked.

Use MIXED DATA

Check this box to show that the DB2 installation option MIXED DATA is set to YES.

Default: Unchecked.

DB2/390 Generate Page

This page enables you to set the defaults DataAtlas uses when DB2/390 table definitions are generated.

Field

Output file

Check Boxes

Append to existing file
Generate object descriptions for DataGuide
Convert names to uppercase

Radio Buttons

Create
Alter
Drop
Drop/Create

Push Buttons

Search
Reset

Output File

This field names the file to which DataAtlas generates output for DB2/390 objects. Specify a full path for the file unless you want it to be in the directory where DataAtlas is running. If so, the file name alone is sufficient.

You can override this entry by specifying a different output file in the **Generate to File** notebook.

Default: A file named 'genmvs.ddl'.

Append to Existing File

Check this box here or in the **Generate to File** notebook to append the generated output to the contents of an existing file. If you do not check this box and the file exists, DataAtlas will overwrite its contents.

Generate Object Descriptions for DataGuide

Check this box if there is descriptive text for each table, and you want DataAtlas to put it into a separate file when it generates the DDL file. The descriptive file will have a name of the form *creator.Tablename* . Because of the length of the name, the output drive must be HPFS or NTFS.

IBM DataGuide can extract text from the descriptive file for its information catalog.

Required?: No.

Default: Unchecked.

Create

Select this button to generate file that will contain CREATE statements. (This DDL generation button is selected by default.)

Alter

Select this button to generate a file that will contain ALTER statements. When it is selected, the **Database Alias** field is enabled.

Drop

Select this button to generate file that will contain DROP statements.

Drop/Create

Select this button to generate file that will contain a DROP and CREATE statement for each object.

Search

Click this button to search for and select an existing file that will contain the generated output.

Reset

Click this button to return the settings on this page to what they were when either the **OK** or **Save** was last clicked.

DB2 UDB Page

This page enables you to set the defaults that DataAtlas uses when DB2 UDB tables are populated.

Fields

- System
- Database
- Schema
- Relational design
- Physical design
- Use mapping table

Push Buttons

- Search
- Clear
- Reset

Check Boxes

- Reconcile
- Use or create data elements for all columns

System

This field contains the name of the server.

Required?: No.

Restriction: The system object must already exist in the TeamConnection database.

Physical Design

This field contains the name of a physical design object with which you want to associate DB2 UDB tables when they are populated.

Required?: No.

Restriction: The physical design object must already exist in the TeamConnection database.

DB2 UDB Generate Page

This page enables you to set the defaults DataAtlas uses when DB2/390 DDL is generated.

Field

- Output file

Check Boxes

- Append to existing file
- Generate object descriptions for DataGuide
- Convert names to uppercase

Radio Buttons

- Alter
- Create
- Drop
- Drop/Create

Push Buttons

- Search
- Reset

Output File

This field names the file to which DataAtlas generates output for DB2 UDB objects. Specify a full path for the file unless you want it to be in the directory where DataAtlas is running. If so, the file name alone is sufficient.

You can override this entry by specifying a different output file in the **Generate to File** notebook.

Default: A file named 'gendb2cs.ddl'.

Alter

Select this button to generate a file that will contain an ALTER statement.

General Page

This page enables you to specify the directories where DataAtlas writes temporary files and query and populate results.

Fields

- File name
- Working directory
- Required field color
- Default HLL
- Query
- Populate
- Validate

Push Buttons

- Search
- Reset

File Name

This is a read-only field that contains the name of the file in which the profile is stored.

Working Directory

This field specifies the complete path (drive and directory) of a location where DataAtlas can write temporary files.

Default: The directory where DataAtlas is installed.

Required Field Color

Select the color to be used for the labels of fields in properties notebooks that must contain an entry.

Default: dark pink.

Default HLL

Select a default high-level language whose objects you are likely to generate. Your selection determines the default settings of the **Generate to File** notebook's **Compile Options** page.

Default: COBOL.

Query

This field specifies the complete path (drive and directory) you want DataAtlas to use when it stores query reports.

Default: The directory where DataAtlas is installed.

Populate

This field specifies the complete path (drive and directory) you want DataAtlas to use to store populate reports and prototype mapping tables.

Default: The directory where DataAtlas resides.

Validate

This field specifies the complete path (drive and directory) you want DataAtlas to use when it stores validation reports. DataAtlas produces a validation report whenever it detects an error in the generation of a DBD, PSB, COBOL COPY file, or PL/I include file.

Default: The directory where DataAtlas is installed.

Oracle Page

This page enables you to set defaults for fields used in defining Oracle objects.

Fields

- System
- Schema
- Relational design
- Physical design

Push Buttons

- Search
- Clear
- Reset

Relational Design

This field contains the name of a relational design object with which you want to associate Oracle objects.

Required?: No.

Restriction: The relational design object must already exist in the TeamConnection database.

Physical Design

This field contains the name of a physical design object with which you want to associate Oracle objects.

Required?: No.

Restriction: The physical design object must already exist in the TeamConnection database.

Oracle Generate Page

This page enables you to set the defaults DataAtlas uses when Oracle DDL is generated.

Field

- Output file

Check Boxes

- Append to existing file
- Generate object descriptions for DataGuide
- Convert names to uppercase

Radio Buttons

- Create
- Drop
- Drop/Create

Push Buttons

- Search
- Reset

Output File

This field names the file to which DataAtlas generates output for Oracle objects. Specify a full path for the file unless you want it to be in the directory where DataAtlas is running. If so, the file name alone is sufficient.

You can override this entry by specifying a different output file in the **Generate to File** notebook.

Default: A file named 'genoracl.ddl'.

IMS Page

This page enables you to set the defaults that DataAtlas uses when IMS DBDs and PSBs are populated.

Fields

- DBD
- PSB
- Use mapping table

Check Box

- Reconcile

Push Buttons

- Search
- Reset

DBD

This field identifies the directory containing the input files for the DBD to be populated.

Required?: No.

PSB

This field identifies the directory containing the input files for the PSB to be populated.

Required?: No.

IMS Generate Page

This pages enables you to set defaults for generating IMS DBD and PSB source files.

Fields

- DBD
- PSB

Check Box

- Generate object description for DataGuide

Push Buttons

- Search
- Reset

Radio Buttons

- Never, On errors, Always

DBD

This field identifies the directory that will contain the generated DBD source file.   The drive must be included with the directory name.

Required?: No.

PSB

This field identifies the directory that will contain the generated PSB source file.   The drive must be included with the directory name.

Required?: No.

Generate Object Descriptions for DataGuide

Check this box if there is descriptive text for the object, and you want DataAtlas to insert it into the generated output as comments.

IBM DataGuide can extract the comments for its information catalog.

Required?: No.

Default: Checked.

Never, On Errors, Always

Select the button that tells DataAtlas when to produce a report of the generate process - never, when an error occurs, or always.

Default: **On errors** .

Physical Design Page

This page enables you to set the defaults DataAtlas uses when all the objects in a physical design are generated.

Field

- Output directory

Check Box

- Generate object descriptions for DataGuide
- Convert names to uppercase

Radio Buttons

- Create
- Drop
- Drop/Create

Push Buttons

- Search
- Reset

Output Directory

This field identifies the directory that will contain the generated DDL. The drive must be included with the directory name.

Required?: No.

Default: The current directory.

PL/I Page

This page enables you to set the defaults that DataAtlas uses when PL/I include files are populated.

Fields

- Compiler options
- Command file
- Source directory
- Use mapping table

Check Boxes

- Use command file not compiler options
- Reconcile

Push Buttons

- Search
- Reset

Compiler Options

This field contains PL/I for OS/2 Version 1.2 compiler options to be used in the compilation phase of populating a PL/I include file.   Alternatively, you can put compiler options in a command file.

In general, you cannot specify compiler options that conflict with this set:

```
XINFO(SYM) FLAG(I) MACRO NOCOMPILE SOURCE ATTRIBUTES(FULL) LANGLVL(SAA NOEXT)
```

For example, you cannot specify XINFO(NOSYM), which says not to generate variable information in the .ADT (SYSDATA) file.

DataAtlas will support a subset of the keywords that are valid when LANGLVL(SAA NOEXT) is specified, such as NONVARYING.

Required?: No.

Command File

This field identifies a file that contains PL/I for OS/2 Version 1.2 compiler options to be used in the compilation phase of populating a PL/I included file. Alternatively, you can specify compiler options in the **Compile options** field.

The **Use options in command file** box enables the **Command file** field.

Required?: No.

Source Directory

This field identifies a directory that contains PL/I include files to be populated.

Required?: No.

Default: The current directory.

PL/I Generate Page

This pages enables you to set defaults for generating PL/I include files.

Fields

- Output directory
- Start level
- Increment

Push Buttons

- Search
- Reset

Radio Buttons

- Never, On errors, Always

Output Directory

This field identifies the directory that will contain generated PL/I included files.   The drive must be included with the directory name.

Required?: No.

Default: The current directory.

Increment

Use the spin buttons to select the increment to be used for the second and succeeding elements in generated files.

Default: 1.

Query Page

This page enables you to make choices that apply to all queries and query reports.

Field

Directory for stored queries

List

Separate fields with

Push Buttons

Search
Reset

Directory for Stored Queries

This field identifies the directory where the files containing queries are to be stored.

Required?: No.

Default: `&lt.DataAtlasInstallPath>\lang\en_us\samples\query` .

`&lt.DataAtlasInstallPath>` is the root directory where DataAtlas is installed.

Separate Fields With

This list contains the characters that can be used to separate the fields in a query report.   Select a character or specify one in the field provided.

Required?: No.

Default: Comma.

TeamConnection Page

This page enables you to modify the version information and name qualifiers that are associated with the objects you store in the TeamConnection database.

Fields

- Family
- Release
- Work area
- Component
- Variation
- Revision

Push Button

- Search
- Reset

Family

In this field, enter the name of a family.

A family is a complete and self-contained collection of TeamConnection users and development data. Data within a family is isolated from data in other families.

Required?: Yes. This field cannot be blank.

Release

This field contains the name of a release.

A release is a logical organization of all the parts related to an application; that is, all the parts that must be built, tested, and distributed together.

Use the **Search** push button to see a list of names that can be entered here.

Required?: Yes. You must enter a release name in the window that appears when DataAtlas starts up for the first time. That name will be transferred to this field. This field can be changed, but it cannot be blanked out and saved.

Limit: 15 characters.

Work Area

This field contains the name of a work area.

A work area is a view of a TeamConnection release. Release parts can be checked out to a work area, updated, and used in builds without affecting the integrated version of the parts in the release.

Use the **Search** push button to see a list of names that can be entered here.

Required?: Yes. You must enter a work area name in the window that appears when DataAtlas starts up for the first time. That name will be transferred to this field. This field can be changed, but it cannot be blanked out and saved.

Limit: 15 characters.

Component

This field contains the name of a component.

A component is a logical group of development data within a TeamConnection family. Identifying a component is a means of controlling access to development data.

Use the **Search** push button to see a list of names that can be entered here.

Required?: Yes. You must enter a component name in the window that appears when DataAtlas starts up for the first time. That name will be transferred to this field. This field can be changed, but it cannot be blanked out and saved.

Limit: 31 characters.

Variation

This field contains the name of a variation.

A variation name becomes part of the full name of an object in the TeamConnection database.   It can be useful in distinguishing between similar objects that have different uses.

If an object is related to a higher-level object, it takes the variation name, if any, of the higher-level object.   So, for example, a relational database object will automatically be assigned the variation name given to the higher-level system object when it was defined.   Further, the same variation name will be assigned to table spaces within that database and to tables within one of the table spaces.   Variation names are propagated in the same way from IMS DBDs to their segments and fields.

Required?: No.

Revision

This field contains the name of a revision.

A revision is a qualifier that becomes part of the full name of an object in the TeamConnection database. This qualifier can be useful in distinguishing between similar objects that have to be available at the same time.

If an object is related to a higher-level object, it takes the revision name, if any, of the higher-level object. So, for example, a relational database object will automatically be assigned the revision name given to the higher-level system object when it was defined. Further, the same revision name will be assigned to table spaces within that database and to tables within one of the table spaces. Revision names are propagated in the same way from IMS DBDs to their segments and fields.

Required?: No.

Search

Click this button to see a list of the releases, work areas, or components represented in the TeamConnection database.   When you select a name from a list, it is put in the appropriate field.

You must have a valid family to search for a release or component.   You must have a valid family and release to search for a work area.

Proposal Report Window

In this window, you see the results of the performed actions for getting design proposals for the selected objects.

The results are presented in terms of proposal report items, as well as information report items. Information report items inform you on what information is missing to generate appropriate proposals. Proposal report items tell you which design step DataAtlas Designer proposes to take.

You can accept or reject a proposal.

On request, you get more information on a proposal report item.

Also, you can look the rule which caused a report item.

Tasks

- Evaluating the design support report
- Executing design proposals

Push Buttons

- Execute
- Cancel
- Help

Menu-Bar Choices

- Report
- Selected
- Edit
- Window
- Help

Pop-up menu

- Accept
- Reject
- Rule
- More

Selected

This choice displays two actions you can apply to the selected proposals:

Accept. Marks the highlighted (selected) proposals as **Accepted** - **Yes** . When you execute proposals , it is the proposals marked **Accepted** - **Yes** that are executed.

Reject. Marks the highlighted (selected) proposals as **Accepted** - **No** .

Accept

This pop-up menu item marks the highlighted (selected) proposals as Accepted - **Yes** .

When you execute proposals , it is the proposals marked **Accepted** - **Yes** that are executed.

Reject

This pop-up menu item marks the highlighted (selected) proposals as Accepted - **No** .

Only proposals marked **Accepted** - **Yes** are executed .

Execute

Click this button to let DataAtlas Designer automatically execute the accepted proposals.

Proposal Report Item Window

A report item reports the result of performing a previously selected action item.

You can request this window during the evaluation of a design support report.

In this window, you see the detailed description of a proposal report item, and the objects affected by the action item.

Tasks

You can look at the notebooks of affected objects by double-clicking on the icons. In addition to the icon of the object currently under design, you also get icons for the proposed objects. The notebooks of the proposed objects already contain the proposals.

Push Buttons

- Cancel
- Help

PSB Notebook

This notebook enables you to create and update the definition of an IMS program specification block (PSB).

Menu-Bar Choices

- PSB
- Window
- Help

Pages

- PSBGEN
- General

Push Buttons

- OK
- Save
- Cancel
- Help

PSB

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

General Page

This page enables you to name and describe in text the PSB you are defining.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the PSB.

Required?: Yes.

Limit: 8 alphanumeric characters.

Requirement: If the program will run in a message processing region, this name must be the same as the program load module name in the IMS.PGMLIB program library.

Variation

This field contains the name of a variation.

A variation name becomes part of the full name of an object in the TeamConnection database. It can be useful in distinguishing between similar objects that have different uses.

If an object is related to a higher-level object, it takes the variation name, if any, of the higher-level object. So, for example, a variation name assigned to a PSB object will automatically be given to the lower-level PCB when it is defined.

Required?: No.

Revision

This field contains the name of a revision.

A revision is a qualifier that becomes part of the full name of an object in the TeamConnection database. This qualifier can be useful in distinguishing between similar objects that have to be available at the same time.

If an object is related to a higher-level object, it takes the revision name, if any, of the higher-level object. So, for example, a revision name assigned to a PSB object will automatically be given to the lower-level PCB when it is defined.

Required?: No.

PSBGEN Page

This page enables you to specify the characteristics of an application program.

Fields

- I/O area size
- SSA size
- I/O error option
- Maximum Qx calls
- Maximum locks

List

- Language

Radio Buttons

- Compatible
- Online Image Copy
- Write to Operator

Table

- PCBs

Push Buttons

- Create
- Search
- Open
- Delete

I/O Area Size

This field specifies the size in bytes of the largest I/O area used by the application program.

Required?: No.

Default: A size calculated by the ACB utility.

Limit: 256000. However, the combined length of all the concatenated segments to be returned to the application on a single path call must not exceed 65535 bytes.

SSA Size

This field specifies the size in bytes of the maximum total length of all the SSAs used by the application program.

Required?: No.

Default: A size calculated by the ACB utility.

Limit: 256000.

I/O Error Option

This field contains a decimal number that is returned to the operating system when IMS terminates normally but an input or output error occurs on a database during the application program's execution.

Required?: Only if the WTOR radio button is selected.

Limits: 0 to 4095.

Maximum Qx Calls

This field specifies the maximum number of database calls with Qx command codes that can be issued between synchronization points.

Required?: No.

Default: 0.

Maximum Locks

This field specifies the maximum number of locks an application program can get at one time.

Required?: No.

Default: 0.

Limits: 0 to 255, but each unit represents 1000 locks. So entering 5 would specify a maximum of 5000 locks at one time.

Language

Select from this list the compiler language in which the message processing or batch processing program is written. Select the blank entry if the application is enabled for the IBM Language Environment for MVS & VM.

If you selected the **OLIC** radio button, do not select **PL/I** from the list. If your application program is written in C language, select **ASSEM** .

Required?: No, you can select the blank entry.

Compatible

Select this button if you want this PSB to be treated as if it has an I/O PCB. Do not select it if the PSB has an I/O PCB added only for BMP and MSG regions.

Default: Deselected.

Online Image Copy

Select this button if you want to authorize the user of this PSB to run the Online Database Image Copy utility or the Surveyor utility against a database named in this PSB.

Default: Deselected.

Restrictions: Do not select this button if:

⇑ Any DBPCB (TYPE=DB) specifies PROCOPT=L or LS.
⇑ The database in question is a CICS/MVS GSAM, HSAM, MSDB, or DEDB database.

Write to Operator

Select this button if you want the DFS0451A error message to be issued when IMS terminates normally but an input or output error occurs on a database during the application program's execution.

Required?: No.

PCBs

This table shows the name and type of each PCBs in this PSB.   Also, it shows the base DBD for each GSAM and DB PCB.

Create

Click this button to create a new PCB and add it to this PSB.

Search

Click this button to search for an existing PCB and add it to this PSB.

Open

Click this button to open the **PCB** notebook for the PCB you have selected from the **PCBs** table.

Delete

Click this button to delete from this PSB the PCB you have selected from the **PCBs** table.

Query File - Search Window

This window enables you to specify and load a file containing a query you want to run. The query must be documented like a supplied query.

Field

Open filename

Lists

Type of file
Drive
Directory
File

Push Buttons

OK
Cancel
Help

Open Filename

This field identifies a file that contains a documented query. You can fill in the field by typing into it directly or by clicking an entry in the **File** list.

When you click **OK**, the name in this field is entered into the **File name** field on the **General** page of the **SQL Query** notebook you came from. The query's name and descriptive text are entered into the **Name** and **Description** fields, respectively. The query itself is written on the notebook's **Query** page.

Required?: Yes.

File

Click a file name in this list and it's entered into the **Open filename** field. Double-click a file name and it's entered into the **File name** field on the **General** page of the **SQL Query** notebook you came from. The query's name and descriptive text are entered into the **Name** and **Description** fields, respectively. The query itself is written on the notebook's **Query** page. Double-clicking a file name also closes this window.

Relational Design Notebook

This notebook contains all the table definitions in a relational design . It's used with DataAtlas Modeler to identify all the tables that participate in a conceptual data model and in one or more physical designs.

In this notebook you can:

⇑ Maintain the list of table definitions associated with a relational design

⇑ Create and update the DB2/390, DB2 UDB, and Oracle physical designs that are associated with a relational design

Menu-Bar Choices

Relational design
Window
Help

Pages

Definitions
DB2/390
DB2 UDB
Oracle
General

Push Buttons

OK
Save
Cancel
Help

Relational Design

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

Definitions Page

This page enables you to maintain a list of shareable table definitions that are associated with this relational design.

Shareable table definitions can be created through a transform from DataAtlas Modeler or within DataAtlas Dictionary.

List

- Shareable Table Definitions

Push Buttons

- Open
- Search
- Remove

Shareable Table Definitions

This list identifies the shareable table definitions that are associated with this relational design. You can add an entry to the list by clicking on the **Search** push button and then selecting a table definition from the search results.

If you select entries in the list, you can use the **Open** push button to open the notebooks for those shareable table definitions. You can use the **Remove** push button to remove entries from the list and from the relational design.

Open

Click this button to open the notebooks of the selected entries in the **Shareable Table Definitions** list.

Search

Click this button to find a shareable table definition object in the TeamConnection database and add it to the **Shareable Table Definitions** list. Adding a shareable table definition to the list associates it with the relational design.

Remove

Click this button to remove the entries you selected from the **Shareable Table Definitions** list and from this relational design. The shareable table definition objects will still exist in the TeamConnection database.

DB2/390 Page

This page enables you to create and update the DB2/390 physical designs that are related to this relational design.

List

- DB2/390 Physical Designs

Push Buttons

- Open
- Create

DB2/390 Physical Designs

This list identifies the DB2/390 physical designs that are associated with this relational design.

If you select an entry in the list, you can use the **Open** push button to open that physical design's properties notebook.

You can use the **Create** push button to create and associate a new physical design with this relational design.

Open

Click this button to open the physical design you selected in the physical designs list.

Create

Click this button to create and associate a new physical design with the relational design.

DB2 UDB Page

This page enables you to create and update the DB2 UDB physical designs that are related to this relational design.

List

DB2 UDB Physical Designs

Push Buttons

Open
Create

DB2 UDB Physical Designs

This list identifies the DB2 UDB physical designs that are associated with this relational design.

If you select an entry in the list, you can use the **Open** push button to open that physical design's properties notebook.

You can use the **Create** push button to create and associate a new physical design with this relational design.

Oracle Page

This page enables you to create and update the Oracle physical designs that are related to this relational design.

List

Oracle Physical Designs

Push Buttons

Open
Create

Oracle Physical Designs

This list identifies the Oracle physical designs that are associated with this relational design.

If you select an entry in the list, you can use the **Open** push button to open that physical design's properties notebook.

You can use the **Create** push button to create and associate a new physical design with this relational design.

General Page

This page enables you to name the relational design and write a description of it.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the relational design.

Required?: Yes.

Relational System Notebook

This notebook enables you to define and modify a relational system object and save it in the TeamConnection database.

Menu-Bar Choices

- System
- Window
- Help

Pages

- System
- General

Push Buttons

- OK
- Save
- Cancel
- Help

System

This choice displays one action, **Save**. Select it to save the settings in this notebook and keep it open.

System Page

This page enables you to:

⇑ Specify the type of the database system
⇑ Add to or modify the list of database creators associated with it.

Lists

Type
Version
Creator/Schema

Check Box

DB2/390 data sharing environment

Push Buttons

Add
Modify

Type

Select the type of the database management system from the list.

Required? Yes.

Restriction: You cannot change the type after another object has been associated with this relational system object.

Version

Select a version of the database management system from the list.

Required? Yes.

Restriction: After you select a version, you can change it only to a later version.

DB2/390 Data Sharing Environment

Check this box if the objects that belong to this system are used in a data sharing environment. You cannot modify this setting after an object that belongs to this system has been stored in the TeamConnection database.

Required? No.

Default: Unchecked.

Creator/Schema

If you choose a DB2/390 system type, this list box is labeled **Creator** . If you choose a DB2 UDB or Oracle system type, it's labeled **Schema** . Use this box to maintain a list of creators or schemas for the corresponding system type.

You can add to the list by:

1. Selecting &lt.NEW> in the list

2. Editing the name in the field below the list

3. Selecting the **Add** push button

You can modify an entry in the list by:

1. Selecting the name of the creator or schema you want to modify

2. Editing the name in the field below the list

3. Selecting the **Modify** push button

A creator/schema name qualifies the names of many relational database object types. Changing it changes the names of the associated objects.

Limit: 8 alphanumeric characters.

Add

Click this button after you have entered the name of a new creator or schema in the field above the list.   The name will be added to the list.

Modify

Click this button to replace the selected creator or schema name with the name you entered in the field above the list.

General Page

This page enables you to name the relational system and write a description of it.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the relational system name - the name under which you group all the tables and associated objects that are related to a given DB2 UDB, DB2/390, or Oracle catalog.

The relational system name is used as the highest-level qualifier name for most relational database object types. If you change this name, you change the names of all the objects within the relational system.

Required?: Yes.

Report Folder

This folder collects all the reports for the SQL queries you have run.   To open the properties notebook for a report, either:

⇑ Double-click its icon or
⇑ Select its icon and click **Settings** under the **Selected** menu-bar choice

Menu-Bar Choices

Folder
Selected
Edit
View
Window
Help

Folder

This choice lets you decide what the folder will look like when it's opened. Its menu has one action, **Open as** , which expands into these options:

Icon view. Shows objects as icons.

Flowed icon view. Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Selected

This choice offers two actions that apply to the selected query report:

Properties. Opens the properties notebook for the selected report.

Original. Leads to a **Delete** option that enables you to delete both the selected icon and the file containing the report.

Edit

This choice offers two actions:

Select all. Highlights (selects) all the reports in the folder.

Deselect all. Removes highlighting from (deselects) all the reports in the folder.

View

This choice controls the appearance and order of the icons in the folder. The items on its pull-down menu are:

Icon view: Shows objects as icons.

Flowed icon view. Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Sort. Offers the **Name** option **,** and which reorders the icons alphabetically by report name.

Arrange. Arranges the icons to fit the size of a resized window.

Rule Window

You can request this window during the evaluation of a design support report.

This window shows the detailed description of the rule on which an action is based.

This rule is applied when the action item is performed, together with the other rules supporting the action.

Save As Window

This window enables you to save the report contents to a file.

Field

Save as filename

Lists

Type of file
Drive
Directory
File

Push Buttons

OK
Cancel
Help

Save As Filename

This field identifies the file in which the report will be saved. Specify the file name you want by:

⇑ Overtyping the file name DataAtlas used in saving the report or

⇑ Clicking a file name in the **File** list

Required?: Yes.

Type of File

Select from the drop-down list a file type to delimit the file names displayed in the **File** list.

Required?: Yes.

Default: All files.

Drive

Select from the drop-down list the drive that will contain or already contains the file you identified in the **Save as filename** field.

Required?: Yes.

Default: The drive of the current directory.

File

This list identifies the files on the selected drive and directory. Click on a file name to enter it into the **Save as filename** field.

Secondary Index Notebook

This notebook enables you to define secondary index relationships.   In it, you can specify:

⇑ The name of an indexed field that is associated with an index target segment type

⇑ The index source segment type

⇑ The index source segment fields that are used in creating a secondary index

Pages

XDFLD
Index Field

Push Buttons

OK
Save
Cancel
Help

XDFLD Page

This pages enables you to specify the details of a secondary index relationship.

Fields

- Segment Name
- DBD Name
- Index DBD
- Index Segment
- XDFLD Name
- Exit Routine
- Constant
- Null value

List

- Source Segment

Check Box

- Symbolic Pointers

Push Button

- Search

Segment Name

This field identifies the index target segment.

Required?: Yes.

DBD Name

This field identifies the DBD that contains the index target segment.

Required?: Yes.

Index DBD

This field identifies the DBD for the index database.

Required?: Yes.

Index Segment

This field identifies the pointer segment in the DBD of the index database.

Required?: Yes.

XDFLD Name

This field contains the name of the indexed data field of an index target segment.

Required?: Yes.

Exit Routine

This field contains the name of an index maintenance exit routine that suppresses the creation of selected index pointer segments. The routine receives control whenever DL/I attempts to insert, delete, or replace an index entry because of changes occurring in the indexed database.

Required?: No.

Constant

This field contains a character with which every index pointer segment in a given secondary index is identified. This character makes it possible to identify all the index pointer segments associated with each secondary index when multiple secondary indexes reside in the same secondary index database.

Required?: No.

Null Value

This field contains a 1-byte term that suppresses the creation of index pointer segments. No indexing is performed when the index source segment data used in the search field of an index pointer segment contains the specified term.

Instead of a 1-byte term, you can specify **BLANK** or **ZERO** . **BLANK** is equivalent to C' '. **ZERO** is equivalent to X'00' or 0, but not C'0'.

Required?: No.

Examples:

```
X'10'
C'Z'
5
X'00101101'
```

Source Segment

From this list, select an index source segment type for this secondary index relationship.

Required?: No. If you make no selection, the index target segment type is assumed to be the index source segment.

Symbolic Pointers

Check this box if you want the concatenated keys of the index target segments, not direct pointers, to be placed in the index pointer segments.

Default: Unchecked.

Index Field Page

This page enables you to specify which fields of the index source segment are search fields and which fields, if any, are subsequence fields and duplicate data fields.

Table

- Fields

Lists

- Search Fields
- Subsequence Fields
- Duplicate Data Fields

Push Buttons

- Add
- Remove

Fields

This table lists all the fields in the index source segment, along with their characteristics.

By selecting a field and using an **Add** push button, you can add the field to the associated list, identifying the field as search field, subsequence field, or duplicate data field.

Search Fields

This list identifies the fields in the index source segment that will be used as the search fields of a secondary index.

Required?: Yes. At least one field must be identified.

Subsequence Fields

This list identifies which, if any, fields in the index source segment will be used as the subsequence fields of a secondary index.

Required?: No.

Duplicate Data Fields

This list identifies which, if any, fields in the index source segment will be used as the duplicate data fields of a secondary index.

Required?: No.

Add

Select one of these push buttons to add a selected field in the **Field** table to the appropriate list.

Remove

Select one of these push buttons to remove a selected field from the associated list.

Segment Notebook (All Segments)

This notebook enables you to create and maintain segment definitions. Each DBD access method has a specific combination segment notebook pages. Only the parameters that are valid for each access method are presented.

Menu-Bar Choices

Segment
Window
Help

Pages

SEGM Page (DEDB)
SEGM Page (HDAM and HIDAM)
SEGM Page (HISAM)
SEGM Page (HSAM)
SEGM Page (Index, SHISAM, and SHSAM)
SEGM Page (Logical)
SEGM Page (MSDB)
COMPRTN Page
EXIT Page
LCHILD Page
LPARENT Page
XDFLD Page
General Page

Push Buttons

OK
Save
Cancel
Help

Segment

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

SEGM Page (DEBD)

This page enables you to specify the attributes and relationships of segments in DEBD databases.

Fields

- Maximum
- Minimum
- Subset Pointers
- Structure name

Lists

- Parent Segment
- Child Pointer
- Placement Rule

Check Box

- Direct Dependent
- Use existing shareable data structure

Table

- Fields

Push Buttons

- Search
- Open (structure)
- Create
- Open (field)
- Delete

Subset Pointers

This field contains a number from 0 to 8 that specifies how many subset pointers you are using. This field is enabled only if the **Direct Dependent** box is checked.

Required?: No. Leaving this field blank has the same meaning as specifying 0.

Placement Rule

Select from the list where new occurrences of the segment type are to be inserted. The selection has meaning only if the segment has no sequence field or a nonunique sequence field. The listed keywords have the following meanings:

FIRST: If the segment has no sequence field, a new occurrence is inserted before all physical twins. If it has a nonunique sequence field, it is inserted before all physical twins with the same sequence field value.

LAST: If the segment has no sequence field, a new occurrence is inserted after all physical twins. If it has a nonunique sequence field, it is inserted after all physical twins with the same sequence field value.

HERE: If the segment has no sequence field, a new occurrence is inserted before the physical twin on which position was established. If a position was not established, the new occurrence is inserted before all physical twins. If it has a nonunique sequence field, a new occurrence is inserted before the physical twin with the same sequence field value on which position was established. If a position was not established, the new occurrence is inserted before all physical twins with the same sequence field value.

Required?: Only when a segment does not have a unique sequence field.

Default: LAST.

Direct Dependent

Check this box if this is a direct-dependent segment type. Do not check it if this is a sequential-dependent segment type.

Restriction: Only one sequential-dependent segment is permitted per DEBD, and it must be the first dependent segment type.

Default: Checked.

Structure Name

If you search for and select a shareable data structure , DataAtlas puts the name of the selected structure in this field.

Use existing shareable data structure

Check this box if you want to use a shareable data structure to define the segment. When you check this box, the **Search** and **Open** push buttons are enabled.

Default: Unchecked.

Search

Click this button to see a list of shareable data structures. If you select a shareable data structure from the list, it is used to define the segment. The name of the shareable data structure will appear in the **Structure name** field.

Open (Structure)

Click this button to open the notebook for the shareable data structure identified in the adjacent **Structure name** field.

SEGM Page (HDAM and HIDAM)

This page enables you to specify the attributes and relationships of segments in HDAM and HIDAM databases.

Fields

- Maximum
- Minimum
- Structure name

Lists

- Pointer
- Parent Segment
- Child Pointer
- Data Sets
- Insert Rule
- Delete Rule
- Replace Rule
- Placement

Check Box

- Use existing shareable data structure

Table

- Fields

Push Buttons

- Search
- Open (structure)
- Create
- Open (field)
- Delete

Maximum

This field specifies the maximum length in bytes of the data portion of the segment.

Required?: Yes.

Minimum

This field specifies the minimum length in bytes of the data portion of the segment.

Required?: Yes, for variable-length segments. The field cannot be used for fixed-length segments.

Pointer

Select from the list the type of pointer field you want to use in the prefix area of the segment. The keywords in the list have the following meanings:

HIER: Reserves a 4-byte hierarchic forward pointer field.
HIERBWD: Reserves a 4-byte forward pointer field and a 4-byte backward pointer field.
TWIN: Reserves a 4-byte physical twin forward pointer field.
TWINBWD: Reserves a 4-byte physical twin forward pointer field and a 4-byte physical twin backward pointer field.
NOTWIN: Prevents space from being reserved for a physical twin forward pointer.

Required?: No.

Parent Segment

Select from the list the name of this segment type's physical parent, or select **0** if this is a root segment.

Required?: Yes.

Child Pointer

Select from the list the type of physical child pointers to be placed in all occurrences of the physical parent of the segment type being defined.

SNGL causes a 4-byte physical child first pointer to be used.

DBLE causes both a 4-byte physical child first and a 4-byte physical child last pointer to be used.

NONE indicates this is a root segment.

Required?: Yes.

Data Sets

Select from the list the name of the data set that contains this segment type.

Required?: Yes.

Insert Rule

Select from the list the path type that must be used to insert a segment. The letters in the list have the following meanings:

P: Physical
L: Logical
V: Virtual

Required?: Only for logical relationships.

Delete Rule

Select from the list the path type that must be used to delete a segment. The letters in the list have the following meanings:

- **P:** Physical
- **L:** Logical
- **V:** Virtual
- **B:** Bidirectional virtual

Required?: Only for logical relationships.

Replace Rule

Select from the list the path type that must be used to replace a segment. The letters in the list have the following meanings:

- **P:** Physical
- **L:** Logical
- **V:** Virtual

Required?: Only for logical relationships.

Placement

Select from the list where new occurrences of the segment type are to be inserted. The selection has meaning only if the segment has no sequence field or a nonunique sequence field. The listed keywords have the following meanings:

FIRST: If the segment has no sequence field, a new occurrence is inserted before all physical twins. If it has a nonunique sequence field, it is inserted before all physical twins with the same sequence field value.

LAST: If the segment has no sequence field, a new occurrence is inserted after all physical twins. If it has a nonunique sequence field, it is inserted after all physical twins with the same sequence field value.

HERE: If the segment has no sequence field, a new occurrence is inserted before the physical twin on which position was established. If a position was not established, the new occurrence is inserted before all physical twins. If it has a nonunique sequence field, a new occurrence is inserted before the physical twin with the same sequence field value on which position was established. If a position was not established, the new occurrence is inserted before all physical twins with the same sequence field value.

Required?: Only when a segment does not have a unique sequence field.

Default: LAST.

Fields

This table lists the fields used in the segment.

The **Open** and **Delete** push buttons work with whatever field you select in the table. The **Create** push button creates a new field in the table.

Create

Click this button to open the **Field** notebook and define a new field for this segment type.

Open (Field)

Click this button to open the **Field** notebook for whatever field is highlighted in the **Field** table.

Delete

Click this button to delete the highlighted field from this segment type.

SEGM Page (HISAM)

This page enables you to specify the attributes and relationships of segments in HISAM databases.

Fields

- Maximum
- Minimum
- Frequency
- Structure name

Lists

- Parent Segment
- Insert Rule
- Delete Rule
- Replace Rule
- Placement

Check Box

- Use existing shareable data structure

Table

- Fields

Push Buttons

- Search
- Open (structure)
- Create
- Open (field)
- Delete

Frequency

This field contains the **estimated** number of times this segment type is likely to occur for each occurrence of its physical parent. If this is a root segment, this field contains the estimate of the maximum number of database records that appear in the database being defined.

Required?: No.

Limits: From 0.01 to 2 to the 24th power minus 1.

SEGM Page (HSAM)

This page enables you to specify the attributes and relationships of segments in HSAM databases.

Fields

- Length
- Frequency
- Structure name

List

- Parent Segment

Check Box

- Use existing shareable data structure

Table

- Fields

Push Buttons

- Search
- Open (structure)
- Create
- Open (field)
- Delete

Length

This field contains the length in bytes of the data portion of this segment type.

Required?: Yes.

SEGM Page (Index, SHISAM, and SHSAM)

This page enables you to specify the attributes and relationships of segments in Index, SHISAM, and SHSAM databases.

Fields

- Length
- Frequency
- Structure name

Check Box

- Use existing shareable data structure

Table

- Fields

Push Buttons

- Search
- Open (structure)
- Create
- Open (field)
- Delete

SEGM Page (Logical)

This page enables you to specify the attributes and relationships of segments in logical databases.

Fields

- Parent Segment
- Root Source DBD
- Segment (First Source)
- Segment (Second Source)
- DBD (First Source)
- DBD (Second Source)

Check Boxes

- Second Source
- Data (First Source)
- Data (Second Source)

Push Button

- Search

Root Source DBD

This field identifies the DBD that contains the root segment.

Required: Yes.

Segment (First Source)

Select from this list the segment you want to use as the first source segment.

Required?: Yes.

Segment (Second Source)

This read-only field identifies the logical parent segment type that can be used as a second source.

Required?: No.

DBD (First Source)

This read-only field identifies the DBD that contains the source segment.

Required?: Yes.

DBD (Second Source)

This read-only field identifies the DBD that contains the segment used as the second source.

Required?: Yes.

Second Source

Check this box to use the logical parent of the first source segment as the second source segment.

Default: Unchecked.

Data (First Source)

Check this box to specify that:

1. The key portion of the first source segment must be placed in the key feedback area.

2. The segment must be placed in the user I/O area when a call is issued to process the logical segment type that represents the first source segment.

Do not check the box if you want to specify #1 but not #2.

Default: Checked.

Data (Second Source)

Check this box to specify that:

1. The key portion of the second source segment must be placed in the key feedback area.

2. The segment must be placed in the user I/O area when a call is issued to process the logical segment type that represents the second source segment.

Do not check the box if you want to specify #1 but not #2.

Default: Checked.

SEGM Page (MSDB)

This page enables you to specify the attributes and relationships of segments in MSDB databases.

Fields

Length
Structure name

Check Box

Use existing shareable data structure

Table

Fields

Push Buttons

Search
Open (structure)
Create
Open (field)
Delete

COMPRTN Page

This page enables you to identify a Segment Edit/Compression exit routine for HDAM, HIDAM, HISAM, and DEDB databases.   This page should not be used for a logical child segment in any database.

Field

Routine Name

Check Boxes

Data Only
INIT

Routine Name

This field identifies the Segment Edit/Compression exit routine.

Required?: Yes.

Limit: 8 alphanumeric characters.

Data Only

Check this box if you want the exit routine to process data fields only. Do not check it if you want it to process all the segment's fields. (This box maps to the DATA and KEY subparameters of the COMPRTN parameter.)

Restriction: Do not use an exit routine to compress or modify data fields that change the position of the sequence field in respect to the start of the segment.

Requirement: This box must be checked if the segment belongs to a DEDB database or is the root segment of a HISAM database.

Default: Checked.

Initialize

Check this box if the exit routine does initialization and termination processing, and therefore needs control immediately after the database (or first area) is opened and immediately after the database (or last area) is closed. (This box maps to the INIT subparameter of the COMPRTN parameter.)

Default: Unchecked.

EXIT page

This page enables you to identify Data Capture exit routines and select data options for them.

The Data Capture exit routines identified here apply only to the segment type that is the subject of this Segment notebook. To apply Data Capture exit routines to all the segments of a physical database, use the **EXIT** page in the **DBD** notebook.

Table

Data Capture Exits

Field

Capture Routine

Check Boxes

Use DBD Exits
Data
Key
Path
Log
Cascade

Push Buttons

Add
Modify
Delete

Use DBD Exits

Select this check box if you want the exit routines shown on the **EXIT** page of the **DBD** notebook to apply to this segment type.

Default: Unchecked.

LCHILD Page

This page lists the logical child relationships defined for the segment type. By selecting an entry in the **Logical Children** table and the appropriate push button, you can add a new logical child relationship, modify an existing one, or delete an existing one.

Table

- Logical Children

Field

- Child DBD

Lists

- Child Segment
- Pair Segment
- Pointer
- Rules

Radio Buttons

- Virtual
- Physical
- LTwin
- LTwin Bwd

Check Boxes

- LParent
- Counter

Push Buttons

- Add
- Modify
- Delete
- Search

Logical Children

This read-only table lists this segment type's logical child relationships.

Child DBD

This field identifies the DBD that defines the logical child segment. You can use the **Search** push button to get a list of DBDs to choose from.

Required?: Yes, unless the DBD is the one identified on the **General** page of the DBD notebook.

Child Segment

Select from this list the name of a logical child segment that you want to associate with the segment type named on the **General** page.

Required?: Yes.

Limit: 8 alphanumeric characters.

Pair Segment

Select from the list a physical child to be paired with a logical child segment selected from the **Child Segment** list.

Required?: Only for bidirectional logical relationships. Otherwise, make no selection from the list.

Add

Click this button to add an entry for a logical child to the **Logical Children** table. You must first define the new logical child by making the appropriate specifications on this page.

Modify

Click this button to modify an entry for a logical child in the **Logical Children** table. You must first select the entry to be modified and then change one or more of the specifications on this page.

Delete

Click this button to delete the logical child entry you highlighted in **Logical Children** table.

LPARENT Page

This page enables you to define the details of a logical relationship. You can access this page by tabbing to it in the **Segment** notebook or by opening a logical child entry on the **LCHILD** page of the **Segment** notebook.

Fields

- Parent DBD

Lists

- Parent Segment
- Pair Segment
- Pointer
- Rules

Radio Buttons

- Virtual
- Physical
- LTwin
- LTwin Bwd

Check Boxes

- LParent
- Counter
- Define Virtual Logical Child

Push Button

- Search

Parent DBD

This field identifies the DBD that defines the logical parent segment. You can use the **Search** button to get a list of DBDs to choose from.

Required?: Yes.

Parent Segment

Select from the list the name of the logical parent segment, or select **0** if this is a root segment.

Required?: Yes.

Pair Segment

Select a segment from this list for pairing only if you have checked the **Define Virtual Logical Child** box, indicating a virtual relationship to a logical parent segment.

Pointer

Select from this list the kind of pointers you want for this logical relationship. The meanings of the listed keywords are as follows:

SNGL: Used for logical relationships implemented with direct address pointers.

DBLE: Used to reserve two 4-byte pointer fields, logical child first and logical child last, in the logical parent segment.

NONE: Used when the relationship from the logical parent to the logical child segment is not implemented or not implemented with direct address logical child pointers.

Required?: No.

Default: SNGL, unless this is a unidirectional or a physically paired bidirectional logical relationship. Then the default is NONE.

Rules

Select a keyword from this list when no sequence field or a nonunique sequence field has been defined for a virtual logical child. Selecting a keyword controls how occurrences of the real logical child are sequenced from the logical parent through logical child and logical twin pointers.

The meanings of the keywords are:

FIRST: If the logical child has no sequence field, a new occurrence is inserted before the first occurrence of the logical child. If it has a nonunique sequence field, a new occurrence is inserted before all occurrences with the same key.

LAST: If the logical child has no sequence field, a new occurrence is inserted after the last occurrence of the logical child. If it has a nonunique sequence field, a new occurrence is inserted after all occurrences with the same keys.

HERE: If the logical child has no sequence field, a new occurrence is inserted before the logical twin on which position was established by the previous call. If a position was not established, the new occurrence is inserted before all logical twins. If it has a nonunique sequence field, a new occurrence is inserted before the logical twin with the same sequence field value on which position was established by the previous call. If a position was not established on a logical twin with the same sequence field value, the new occurrence is inserted before all twins with the same sequence field value.

Default: LAST.

Virtual

Select this button if you want only intersection data (as opposed to a symbolic pointer to the logical parent) stored with each logical child segment.

Default: Selected.

Physical

Select this button if you want the concatenated key of the logical parent stored with each logical child segment.

You must select this button if:

⇑ The logical parent is in a HISAM database
⇑ A logical child segment is sequenced on its physical twin chain through the use of any part of the logical parent's concatenated key

Default: Deselected.

LTwin

Select this button for virtually paired logical relationships only when defining a real logical child.

A field for a 4-byte logical twin forward pointer is reserved in the prefix of occurrences of the logical child segment type being defined.

Restriction: This button applies only to a logical child being defined in an HDAM or HIDAM database.

LTwin Bwd

Select this button for virtually paired logical relationships only when defining a real logical child.

Fields for a 4-byte logical twin forward pointer and for a 4-byte logical twin backward pointer are reserved in the prefix of occurrences of the logical child segment type being defined.

Restriction: This button applies only to a logical child being defined in an HDAM or HIDAM database.

LParent

Check this box to reserve a 4-byte logical parent pointer field in the prefix of occurrences of the logical child segment type being defined.

Restriction: This button applies only to a logical child being defined in an HDAM or HIDAM database.

Counter

Check this box to reserve a 4-byte counter field in the prefix of occurrences of the logical child segment type being defined.

Required?: Only if a logical parent segment in a HISAM, HDAM, or HIDAM database has logical child segments that are not connected to it by logical child pointers.

Define Virtual Logical Child

Check this box to specify a virtually paired relationship.   The specified pair segment will be a virtual logical child.

If you do not check this box, DataAtlas creates a bidirectional physical relationship.   The paired logical child relationship is automatically created when this DBD is stored.

Required?: No.

Search

Click this button to see a list of DBDs represented in the TeamConnection database. When you select a DBD from the list, its name is returned to the field adjacent to this button.

XDFLD Page

This page lists the secondary index relationships defined for the segment type. By selecting a row in the **Secondary Indexes** table and the appropriate push button, you can create a new secondary index relationship, modify an existing one, or delete an existing one.

Table

- Secondary Indexes

Push Buttons

- Create
- Open
- Delete

Secondary Indexes

This read-only table lists this segment type's index relationships. For each entry, there are corresponding entry fields on the **XDFLD** page of the **Secondary Indexes** notebook.

Create

Click this button to go to the **XDFLD** page of the **Secondary Indexes** notebook and define a new index relationship for this segment type.

Open

Click this button to open the **XDFLD** page of the **Secondary Indexes** notebook and modify the information for the relationship you have highlighted in the **Secondary Indexes** table.

Delete

Click this button to delete the index relationship that you have highlighted in the **Secondary Indexes** table.

General Page

This page enables you to name and describe in text the segment type you are defining.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the segment type.

Required?: Yes.

Limit: 8 alphanumeric characters.

Description

This field contains a text description of the object.

Required?: No.

History

This read-only field shows, by date and time, when the object was created and when it was last modified.

Search

Click this button to see a list of components represented in the TeamConnection database. When you select a component name from the list, its name is put in the **Component** field.

Sensitive Segment Notebook

This notebooks enables you to define a hierarchically related set of data segments to which a program, through its PCB, is sensitive.

Menu-Bar Choices

- Sensitive segment
- Window
- Help

Pages

- SENSEG
- SENFIELD

Push Buttons

- OK
- Save
- Cancel
- Help

Sensitive Segment

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

SENFLD Page

This page enables you to specify the fields within a segment to which an application program is sensitive.

Table

Sensitive Fields

List

Field Name

Field

Start Position

Check Box

Replace

Push Buttons

Add
Modify
Delete

Sensitive Fields

This table lists the fields in the segment to which an application program is sensitive.

Add fields to the table by:

1. Selecting a field from the **Field Name** list
2. Specifying its start position in the **Start Position** field
3. Checking the **Replace** box if appropriate
4. Selecting the **Add** push button

Modify an entry in the table by selecting it, performing one or more of steps 1 to 3 above, and selecting the **Modify** push button.

Delete an entry from the table by selecting the entry and the **Delete** push button.

Field Name

This list contains the names of the fields in the segment you identified as sensitive on the **SENSEG** page. Select one that you want to either add to the **Sensitive Fields** table (**Add** push button) or use in place of a field already in the table (**Modify** push button).

Required?: Yes.

Start Position

This field contains an integer that specifies the starting position of the sensitive field relative to the beginning of the segment within the user's I/O area. The starting position for the first byte of a segment is 1.

Required?: Yes.

Limit: 32767.

Replace

Check this box to allow this sensitive field to be altered by a Replace call.

Default: Checked.

Add

Click this button to add a field you selected from the **Field Name** list, with your **Start Position** and **Replace** specifications, to the **Sensitive Fields** table.

Modify

Click this button to modify a selected entry in the **Sensitive Fields** table. The table is modified to reflect your **Field Name** , **Start Position** , and **Replace** specifications.

Delete

Click this button to delete a selected entry from the **Sensitive Fields** table.

SENSEG Page

This page enables you to name and specify the characteristics of a sensitive segment.

Lists

- Segment Name
- Segment Secondary Indexes
- Indices

Fields

- Parent Segment
- Processing Options

Radio Buttons

- Subset Pointer Sensitivity

Segment Name

This list contains the names of the segment types in the base DBD.   Select the one that is the sensitive segment you want to describe.

Required?: Yes.

Segment Secondary Indexes

This list contains the names secondary indexes that can be used to index the sensitive segment. If you want the segment to be indexed, select a secondary index by double clicking on it. This will move it to the **Indices** list. To move more than one secondary index to the **Indices** list:

1. Click them individually while holding down the Ctrl key or drag the mouse pointer through the list.

2. Click **>** .

Required?: No.

Parent Segment

This field contains the name of this segment's parent segment.   If this segment is a root segment, this field contains a 0.

Required?: Only for dependent segments.

Default: 0.

Indices

This list contains the secondary indexes you selected from the **Segment Secondary Indexes** list.

You can deselect a secondary index in this list by double-clicking on it. This will return it to the **Segment Secondary Indexes** list. To return more than one secondary index to the **Segment Secondary Indexes** list:

1. Click them individually while holding down the Ctrl key or drag the mouse pointer through the list.

2. Click **&lt.** .

Processing Options

This field specifies the processing options that an associated application program can use with this sensitive segment.

All the processing options that you can specify in the Processing Options field in a PCB notebook can be specified in this field. In addition, you can specify the **K** option here. It indicates key-sensitivity only. A GN call with no SSAs can access only data-sensitive segments. If a key-sensitive segment is designated for retrieval in an SSA, the segment is not moved to the user's I/O area. The key is placed at the appropriate offset in the key feedback area of the PCB.

Restriction: If you specified the **L** or **LS** processing option in the **PCB** notebook, leave this field blank.

Subset Pointer Sensitivity

These radio buttons specify the kind of sensitivity you want to assign to each of up to eight subset pointers. Select **R** for read sensitivity, **U** for update sensitivity, or **N** for no sensitivity.

Required?: No.

Default: **R** .

Restriction: Do not select **U** with a processing option of **GO** .

Shareable Data Element Notebook

This notebook enables you to define the characteristics of a shareable data element (SDE).

Menu-Bar Choices

- Element
- Window
- Help

Pages

- Definition
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Element

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

Definition Page

This page enables you to define a shareable data element (SDE) in terms that are appropriate for DB2, Oracle, and IMS databases; VisualAge Generator; COBOL; and PL/I.

The main elements on the page are the **Representation** list and the **Details** group. Each selection from the **Representation** list displays items in the **Details** group that are appropriate to the selection.

List

- Representation

Group

- Details (DB2)
- Details (Oracle)
- Details (IMS)
- Details (VisualAge Generator)
- Details (COBOL)
- Details (PL/I)

Representation

Select from this list the technology that will govern the definition of the SDE. The choices are **DB2** , **Oracle** , **IMS** , **VisualAge Generator** , **COBOL** , and **PL/I** . Your selection determines which items are displayed in the **Details** group.

Default: **DB2** .

Details (DB2)

When you select DB2 from the **Representation** list, the following items are displayed in the **Details** group:

Type. Select a column type for the SDE from this list. The remaining items in the **Details** group will be enabled or disabled based on your selection. The default is CHAR.

Length. Enter the length of the column in this field. The number you enter represents:

⇑ Bytes for CHAR, VARCHAR, DATE, TIME, and TIMESTAMP.

⇑ Double bytes for GRAPHIC and VARGRAPHIC.

⇑ Digits for zoned decimal and floating point numbers (special types of CHAR).

⇑ Kilobytes, megabytes, or gigabytes for CLOB, BLOB, and DBCLOB. Use the spin button at the end of the **Length** field to specify which of these unit sizes is appropriate.

The limits of the number are 1-254 for CHAR, 1-4000 for VARCHAR, 1-127 for GRAPHIC, and 1-2000 for VARGRAPHIC. An entry is required for CHAR, GRAPHIC, VARCHAR, and VARGRAPHIC.

Precision. Enter the precision of the column in this field. The number you enter represents digits for DECIMAL, NUMERIC, and FLOAT, and bits for REAL and DOUBLE. The limits are 1-31 for DECIMAL and NUMERIC, and 1-52 for FLOAT. If you enter 21 or 53, the type is changed to REAL or DOUBLE, respectively, on subsequent viewings of the relational representation. An entry is required for DECIMAL, FLOAT, and NUMERIC.

Scale. Enter the scale of the column in this field. The number you enter represents digits. No entry is required, but if there is one, it must be less than or equal to the precision.

Bit Data. Check this box if the data in the column will be composed of bits. A check is valid only for CHAR, VARCHAR, and LONG VARCHAR.

Details (Oracle)

When you select Oracle from the **Representation** list, the following items are displayed in the **Details** group:

Type. Select a column type for the SDE from this list. The remaining items in the **Details** group will be enabled or disabled based on your selection. The default is CHAR.

Length. Enter the length in bytes of a CHAR (255 maximum), VARCHAR2 (2000 maximum), or RAW (255 maximum) data type. The default length is 1 byte.

Precision. This field applies only to the NUMBER data type. Use the adjacent spin button to specify a precision of 1 to 38 for a fixed-point number. Specify a precision of 0 only for a floating-point number; a precision of 38 is implied even though the field contains a 0.

Scale. This field applies only to the NUMBER data type. Use the adjacent spin button to specify the position of the decimal or binary point relative to the rightmost digit.

Details (IMS)

When you select IMS from the **Representation** list, the following items are displayed in the **Details** group:

Type. Select a field type for the SDE from this list. The default is Character.

Bytes. Enter the length of the field in bytes here. An entry is required. The default is 1 for Character, Hexadecimal, and Packed Decimal; 2 for Halfword; and 4 for Fullword.

Details (VisualAge Generator)

When you select VisualAge Generator from the **Representation** list, the following items are displayed in the **Details** group:

Type. Select an data type for the SDE from this list. The default data type is Character.

Length/Bytes. Select one of these buttons, and enter the element's length or the number of bytes it requires in the adjacent field. An entry in the field is required. The default is 1. The allowable value varies depending on the element type.

Decimals. If the element type is BINARY, NUM, NUMC, PACK, or PACF, select a number here that indicates the number of places to the right of the decimal point. The default is 0.

If you selected the **Length** button, the number in the adjacent field must be greater than or equal to the number you selected for **Decimals** .

Details (COBOL)

When you select COBOL from the **Representation** list, the following items are displayed in the **Details** group:

USAGE. Select a COBOL USAGE clause value from this list. The default value is DISPLAY.

PICTURE. Enter the COBOL PICTURE clause value. An entry is required for BINARY, DISPLAY, DISPLAY-1, and PACKED-DECIMAL usage clauses. The default depends on the COBOL USAGE clause.

SIGN. Select the COBOL SIGN clause value, or leave this field blank. This field is enabled only for DISPLAY USAGE clauses when the PICTURE clause contains a representation for a signed numeric number type; for example, an S and no X, A, or G.

Details (PL/I)

When you select PL/I from the **Representation** list, the following items are displayed in the **Details** group:

Type. Select a data type for the SDE from this list. The default data type is CHARACTER. The **SIGNED** and **UNSIGNED** radio buttons apply only to the FIXED BINARY data type. If **SIGNED** is selected, the fixed binary variable can have a negative sign. The **REAL** and **COMPLEX** radio buttons apply only to the FIXED BINARY, FIXED DECIMAL, FLOAT BINARY, FLOAT DECIMAL, and PICTURE data types.

Precision. Use the adjacent spin button to specify the number of digits that FIXED BINARY, FIXED DECIMAL, FLOAT BINARY, or FLOAT DECIMAL data can have.

Scale. This field applies only to the FIXED BINARY and FIXED DECIMAL data types. Use the adjacent spin button to specify the position of the decimal or binary point relative to the rightmost digit.

Length. This field specifies the length in bytes of CHARACTER, BIT, or GRAPHIC data. The default is 1 byte.

These spin buttons give further information about length:

⇑ **VARYING** says that the length of the data can vary; **NONVARYING** says it can't.

⇑ **VARYINGZ** says that the length of CHARACTER or GRAPHIC data can vary, with a zero byte at the end.

PICTURE. This field contains a picture specification when the data type is PICTURE.

General Page

This page enables you to name the shareable data element (SDE) and write a description of it.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the SDE.

Required?: Yes.

Default: EnterDataElementName.

Shareable Data Structure Notebook

This notebook enables you to define the characteristics of a shareable data structure (SDS).

Menu-Bar Choices

Structure
Window
Help

Pages

General
Tree View

Push Buttons

OK
Save
Cancel
Help

Structure

This choice displays one action, **Save** . Select it to save the settings in this notebook and keep it open.

General Page

This page enables you to name the shareable data structure (SDS) and write a description of it.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the access name of the shareable data structure (SDS).

Required?: Yes.

Tree View Page

This page enables you to define or redefine the objects nested in a shareable data structure (SDS). It's divided into three parts: the **Representation** list, the *tree view* (left-hand side), and the details area (right-hand side), which changes according to the representation and the object you select from the tree view. You use the details area to define the selected object.

The tree view consists of a *root* and its *children* (objects nested within the root).

A child can be a local data element, shareable data element, local data structure, or another SDS. The details in the right-hand area change according to the object you select on the left. You can vary these details to produce the object definition you want.

List

Representation

Tree View

Push Buttons:

Add sibling
Add child
Remove

Radio Buttons:

Elementary
Group
Level 88 (COBOL only)

Details Area

Search box:

Use existing shareable data element/structure
Shareable data element/structure
Search
Open

COBOL details:

Name
REDEFINES
USAGE
PICTURE
SIGN
VALUE
JUSTIFIED
SYNC
BLANK WHEN ZERO
OCCURS (check box)

Additional COBOL details when **OCCURS** *box is checked:*

OCCURS and To (spin buttons)
DEPENDING ON
Available data items
Key data items
Ascending, Descending
INDEXED BY

PL/I details:

Name
DEFINED
POSITION
Type
Precision
Scale
Length
PICTURE
INITIAL
ALIGNED
Arrayed

Additional PL/I details when **ARRAYED** *box is checked:*

Array dimensions
Add
Edit
Remove

IMS details:

Name
REDEFINES
Type
Bytes

Representation

Select **COBOL** , **PL/I** , or **IMS** . If you select **COBOL** or **PL/I** , the controls on the right side of the page will be appropriate for defining the data items in a COBOL or PL/I data structure. If you select **IMS** , the controls will be appropriate for defining the fields in an IMS segment.

Add Child

Click this button to add a child to the tree. The placement of the child depends on the object in the tree that's selected. If it's the root, DataAtlas adds the child either directly below the root (if a level 88) or to the bottom of the tree (all others). If it's a local data component or shareable data component, DataAtlas lets you nest a level-88 object under it.

You can add a child to a selected group object if it is a local data structure. You can add either elementary children or group children if further levels of nesting are needed. Up to 49 levels of nesting are allowed. To add children to a nested shareable data structure, open the shareable data structure's notebook and add the children there.

Elementary

Select this button to add an elementary object to the tree, either as a child of the root, as a child of a local data structure, or as a sibling of a selected child.

An elementary object can be defined by a shareable data element.

Group

Select this button to add a group object to the tree, either as a child of the root or as a sibling of a selected child.

You can define a group object as either a local or shareable data structure. You can update a shareable data structure by opening its notebook and updating it there.

Level 88 (COBOL only)

Select this button to add a level-88 object as either a child or sibling of a selected object other than the root.

A level-88 object adds information about conditional values to its parent object.

Use Existing Shareable Data Element/Structure

This box asks whether you want to use an existing SDE or SDS to define the selected object. If you select an elemental object, the box refers to an SDE; if you select a group object, it refers to an SDS. When you check the box, the **Search** push button is enabled.

Default: Unchecked.

Name

This field contains the COBOL name of the selected object. If you use a shareable data component to define an object in the tree, DataAtlas may put the name of the component in this field. Editing the name doesn't rename the shareable data component; instead the new name becomes an alias for the component.

Multiple objects can be named 'FILLER'. Otherwise, unique names are required for each nested object. If you change the name of an object named 'FILLER', you automatically change its name in the IMS representation too.

Required?: No.

Default: 'FILLER'.

Note: If the **Name** field is empty when the shareable data structure is stored, the name is stored as 'FILLER'.

Type

Select from this list the term that describes the type of data contained in this IMS field.

Default: Character .

OCCURS (Check Box)

Check this box if the selected object in the tree is a table element that can be referred to by indexing or subscripting.

OCCURS and To (Spin Buttons)

Select a value in the **OCCURS** spin button to show the minimum number of times the selected object occurs. Select a value in the **To** spin button to show the maximum number of time it occurs if the **DEPENDING ON** field is filled in. If the **DEPENDING ON** field is empty or if the maximum number is less than or equal to the minimum number, the maximum number is ignored.

Required?: Only for variable-length tables.

DEPENDING ON

Select from this list the name of an object in the tree whose contents specify how many times the selected object occurs in a variable-length table.

Required?: Only for variable-length tables.

Available Data Items

This list identifies the objects you can use as keys. To select an object as a key, either double-click it or click it once and then click **>**. These actions move the name of the object to the **Key data items** list.

Key Data Items

This list identifies objects that are used as keys. To specify that an object is no longer a key, either double- click it or click it once and then click **&lt.** . These actions move the name of the object to the **Available data items** list.

Ascending, Descending

Select the **Ascending** button if the selected entry in the **Key Data Items** list is an ascending key. Select the **Descending** button if it is a descending key.

INDEXED BY

This field identifies the indexes that can be used with the table.

Required?: Only if indexing is used to refer to the selected object.

Name

This field contains the PL/I name of the selected object. If you use a shareable data component to define an object in the tree, DataAtlas may put the name of the component in this field. Editing the name doesn't rename the shareable data component; instead the new name becomes an alias for the component.

Required?: No.

Default: '*'.

DEFINED

Check this box if you want to redefine another object in the tree in terms of the current object. Then enter the name of the object in the adjacent field.

Limitation: **DEFINED** specifications are not carried forward when you populate a PL/I include file. However, what you specify here will be used when you generate an include file.

POSITION

This field is enabled only if the **DEFINED** box is checked and is used only for string-overlay defining. Specify in it the position at which a defined variable begins in a base variable.

This example shows the use of the **POSITION** field:

```
C CHARACTER(10),
X CHARACTER(4) POSITION(7) DEFINED(C);
```

It says that the X character string begins at the 7th character in the C character string and consists of its 4 rightmost characters. To represent the position of X in the **POSITION** field, you would enter '7'.

Required?: No.

Default: None.

Type

Select a data type from this list. The default data type is CHARACTER. The **SIGNED** and **UNSIGNED** radio buttons apply only to the FIXED BIN data type. If **SIGNED** is selected, the fixed binary variable can have a negative sign. The **REAL** and **COMPLEX** radio buttons apply only to the FIXED BIN, FIXED DEC, FLOAT BIN, FLOAT DEC, and PICTURE data types.

Precision

Use the adjacent spin button to specify the number of digits that FIXED BIN, FIXED DEC, FLOAT BIN, or FLOAT DEC data can have.

Scale

This field applies only to the FIXED BIN and FIXED DEC data types. Use the adjacent spin button to specify the position of the decimal or binary point relative to the rightmost digit.

Length

This field specifies the length in bytes of CHARACTER, BIT, or GRAPHIC data. The default is 1 byte.

These radio buttons give further information about length:

- **VARYING** says that the length of the data can vary; **NONVARYING** says it can't.
- **VARYINGZ** says that the length of CHARACTER or GRAPHIC data can vary, with a zero byte at the end.

PICTURE

This field contains a picture specification when the data type is PICTURE.

ALIGNED

Check this box to specify that the selected element, structure, or array is aligned on a storage boundary corresponding to the requirements of its data type.

This box is checked by default for all but the BIT, CHARACTER, GRAPHIC, and PICTURE data types.

Arrayed

Check this box to see the **Array dimensions** table and push buttons used to modify it.

Array Dimensions

This table shows the selected object's array definition. Each row identifies a dimension and its lower and upper bounds. (REFER variables are not yet supported for either bound.)

The table will contain at least 1 dimension if an OCCURS clause exists for the COBOL representation of the selected object. Specifying multiple dimensions from PL/I will cause nested structures with the unchangeable name FILLER to be generated in the COBOL representation. In the IMS representation, an arrayed object will appear as a collapsed object whose length is the sum of the arrayed object's bytes.

Add

Click this button to open the **PL/I Array Dimensions** window and define a new dimension.

Edit

Click this button to open the **PL/I Array Dimension** window and edit the definition of the currently selected dimension.

Remove

Click this button to remove the currently selected dimension and renumber the remaining dimensions.

Name

This field contains the IMS name of the selected object. If you use a shareable data component to define an object in the tree, DataAtlas may put the name of the component in this field. Editing the name doesn't rename the shareable data component; instead the new name becomes an alias for the component.

Multiple objects can have null names. Otherwise, unique names are required for each nested object. The name 'FILLER' is converted to a null name (empty string) when the object is stored.

Required?: No.

Default: 'FILLER'.

REDEFINES

Check this box if you want to redefine the previous object in the tree in terms of the current object.

When you check the box, the name of the object that can be redefined is filled in. If the REDEFINES attribute was set in the PL/I representation, the field may contain a different object name that may not be valid in IMS terms. To reset the value to the valid IMS REDEFINES object, uncheck and recheck the box.

Note: You can't check this box for the first object in the tree in an IMS representation, because it has no preceding object.

Bytes

This field specifies the number of bytes the selected object occupies in the structure.

Shareable Table Definition Notebook

This notebook enables you to view and update a shareable table definition object.

Menu-Bar Choices

- Shareable Table Definition
- Window
- Help

Pages

- Definition
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Shareable Table Definition

This choice displays one action, **Save** . Select it to save the settings and keep the notebook open.

Definition Page

This page enables you to:

⇑ View the relational design and the list of tables that uses this shareable table definition

⇑ Select a new relational design to associate with it

Relational Design

Relational Design

List

Tables

Push Buttons

Search
Open (for Relational Design)
Open (for Tables)

Relational Design

This field contains the name of the DB2 relational design object with which you want to associate this shareable table definition.

To find a DB2 relational design object to associate with this shareable table definition, use the Search push button.

To open the properties notebook for the DB2 relational design object named in this field, use the Open push button.

Required?: No.

Tables

This list identifies tables that:

⇑ Use this shareable table definition
⇑ Belong to the relational design identified in the Relational Design field

To open the properties notebook for a listed table, select the table and the adjacent Open push button.

Requirement: A table must belong to a relational design to be used by DataAtlas Modeler.

Search

Click this button to find a DB2 relational design object to associate with this shareable table definition.

Open (for Relational Design)

Click this button to open the properties notebook for the DB2 relational design object named in the adjacent field.

Open (for Tables)

Click this button to open the properties notebook for a table you selected in the Tables list.

General Page

This page enables you to view and update the description of a shareable table definition.

Fields

- Name
- Variation
- Revision
- Component
- Description
- History

Push Button

- Search

Name

This field contains the system name of the shareable table definition.

Required?: Yes.

SQL Query Folder

This folder contains your SQL queries and the supplied queries that come with DataAtlas. In this folder, you can:

⇑ Create queries
⇑ Make copies of queries
⇑ Delete queries from the folder and, optionally, from the file where they reside

Double-clicking on a query in this folder opens its **SQL Query** notebook.

Menu-Bar Choices

Folder
Selected
Edit
View
Window
Help

Folder

This choice displays two actions:

Open as. Expands into these options...

- ⇑ **Icon view.** Shows objects as icons.
- ⇑ **Flowed icon view.** Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Create SQL query. Adds a new query icon to the folder and opens an **SQL Query** notebook. In the notebook, you can write the query, name it, run it, and save it.

Selected

This choice displays a pull-down menu of actions that apply to the selected SQL query.   The actions are:

Properties. Opens the properties notebook for the selected query.

Delete. Deletes the selected query from the folder.

Original. Expands into these options...

⇑ **Copy.** Creates a new query icon and opens a notebook whose settings are the same as those of the selected query.   You can then name the new query and modify the rest of the notebook as needed.

⇑ **Delete.** Deletes both the selected icon from the folder and the file containing the query.

Edit

This choice offers two selection actions:

Select all. Highlights (selects) all the queries in the folder.

Deselect all. Removes highlighting from (deselects) all the queries in the folder.

View

This choice controls the appearance and order of the icons in the folder. The actions on its menu are:

Icon view. Shows objects as icons.

Flowed icon view. Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Sort. Offers the **Name** option, which reorders the icons alphabetically by object name.

Arrange. Arranges the icons to fit the size of a resized window.

SQL Query Notebook

This notebook enables you to create and save an SQL query, document it with descriptive text, and run it.

Menu-Bar Choices

- Query
- Edit
- View
- Window
- Help

You can use these choices only with the Query Page.

Pages

- General
- Query
- TeamConnection

Push Buttons

- OK
- Save
- Run
- Cancel
- Help

Query

This choice displays one action, **Run**. Select it to execute a query on the **Query** page against the TeamConnection database.

If the query runs successfully, an **SQL Report** notebook opens, displaying the results.

Edit

This choice displays a menu of actions you can use to create or edit an SQL query on the **Query** Page. The actions are:

Import file. Opens the **Import File** window. In it you can identify a file whose contents are imported to the **Query** page. If text already exists on the page, the file contents are appended to the text.

Undo. Resets the contents of the **Query** page to what they were when you last saved them with the **OK** or **Save** buttons.

Cut. Moves whatever is highlighted to the clipboard.

Copy. Copies whatever is highlighted to the clipboard.

Paste. Pastes the contents of the clipboard on the **Query** page.

Clear. Removes whatever is highlighted.

Select all. Highlights (selects) all the text on the **Query** page.

Deselect all. Removes highlighting from (deselects) all the text on the **Query** page.

View

This choice lets you select a word wrap setting for the **Query** page:

Word wrap on. Reflows to the next line any text that extends beyond the editing area.

Word wrap off. Does not reflow text within the editing area.

Default: **Off** .

General Page

This page enables you to name or rename an SQL query and document it with descriptive text.

Fields

- Name
- File Name
- Description
- History

Push Button

- Search

Name

This field specifies a name for the SQL query.

Required?: Yes.

Default: EnterQueryName.

File Name

This field has two uses:

⇑ If you want to copy an existing query from a workstation directory into this notebook, you select the **Search** push button, find the file that contains the query, and select the file. This field will then contain that file's name.

⇑ If you want to store the query on the **Query** page in a workstation file, you enter the file name here and select either the **OK** or **Save** button. If you do not specify a full path, the current directory is used.

Required?: No.

Default: EnterQueryName.qry

Description

This field contains a text description of the SQL query.

Required?: No.

History

This read-only field shows, by date and time, when the SQL query was last modified.

Search

Click this button to find a file that contains a query you want to copy into this notebook. The query must be in the format of a supplied query; that is, statements containing the query's name and descriptive text must precede the SELECT statement.

Query Page

This page is a multiple-line field in which you can type the query.   It must be a valid SQL statement.

The menu-bar choices apply only to this page.

TeamConnection Page

This page enables you to specify a family, release, and work area that will delimit the objects in the TeamConnection database against which your query runs.

Fields

- Family
- Release
- Work Area

Run

Click this button to run the SQL query on the **Query** page against the TeamConnection database.

If the query runs successfully, an **SQL Report** notebook opens, displaying the results.

SQL Report Notebook

This notebook enables you to see the results of an SQL query and store the results in a workstation file.   It also shows you a read-only copy of the query that produced the results.

Menu-Bar Choices

- Report
- Edit
- View
- Window
- Help

Pages

- Report
- Query
- General

Push Buttons

- OK
- Save
- Cancel
- Help

Report

This choice displays one action, **Export to file**. Select it to open the **Export to File** window, in which you specify a workstation file that will contain the report, stripped of descriptive remarks, in ASCII delimited text. ASCII delimited text can be imported into many applications.

Edit

This choice displays a pull-down menu of actions you can perform on the text of the report. The actions are:

Export selected to file. Opens the **Export Selected to File** window, in which you specify a workstation file that will contain selected rows of the report in ASCII delimited text. ASCII delimited text can be imported into many applications.

Copy. Copies selected rows to the clipboard.

Select all. Highlights (selects) the entire report.

Deselect all. Removes highlighting from (deselects) whatever is highlighted in the report.

View

This choice contains the action **Sort columns by name**. Select it to sort the report columns in ascending alphabetic order, left to right.

Report Page

This page contains the query report - that is, the results of the query you ran. You may want to select rows to export to a file or copy to the clipboard.

Query Page

This page contains a read-only copy of the query that generated the report.

General Page

This page enables you to name the report, store it in a workstation file, and write a description of it.

Fields

- Name
- File Name
- Description
- History

Name

This field contains a name for the report.

Required?: No.

Default: EnterReportName.

File Name

This field identifies the workstation file in which the report is stored. The report is stored there only if you select the **OK** or **Save** button.

Required?: No.

Default: The name in the **Name** field, with the extension 'rpt'.

Description

This field contains a text description of the report.

Required?: No.

History

This read-only field shows, by date and time, when the report file was last modified.

Store Window

This page enables you to:

⇑ Include remarks when you store objects into the TeamConnection database

⇑ Store objects into other releases and work areas

⇑ Break a common link

Fields

- Remarks
- Release
- Work Area

Check Box

- Break common link

Push Buttons

- Store
- Cancel
- Help

Remarks

This field contains a text comment about the stored object.

Required?: No.

Release

This read-only field identifies the release into which you want to store the object.

To link parts across releases and work areas, use the Link Parts window in the Team Connection graphical user interface.

If a part is already linked to multiple releases or work areas, you must check Break common link .

Default: The current release is used.

Work Area

This read-only field identifies the work area into which you want to store the object.

To link parts across releases and work areas, use the Link Parts window in the Team Connection graphical user interface.

If a part is already linked to multiple releases or work areas, you must check Break common link .

Default: The current work area is used.

Break Common Link

Check this box if objects in two work areas are linked, or if one work area is refreshed from another work area, and you want to break the link.

In the **Release** and **Work Area** fields, identify only the release and work area into which you want to store the object. Do not identify the release and work area with which you are breaking the link.

Default: Unchecked.

Store

Click this button to store the object in the TeamConnection database.

Table Partition - Data Load Window

Use this window to specify or change the data load for a selected partition.

The data load specifications are estimates which serve as the input for the space calculations for PRIQTY and SECQTY, and for the specification of the ERASE option for table space and index.

If no or insufficient information is entered here, it is not possible to provide reasonable proposals. In this case default values are proposed.

Changing data load values after a proposal has been requested requires to request a new proposal.

Fields

The data load is specified for each partition. The partitions are identified by number. If no partitions are specified, the partition covers the entire table.

⇑ Confidence
⇑ Maintenance Period
⇑ Initial Number of Rows
⇑ Security
⇑ Growth
- Rate
- Period

⇑ Delete
- Rate
- Period

Push Buttons

OK
Save
Reset
Cancel
Help

Table Partition - Work Load Window

Use this dialog to specify or change the work load for a selected partition.

Together with the data load values, the work load values determine the adequate configuration of a table's access and storage structures. DataAtlas Designer uses the workload information for index proposals and free space calculations. If no information is entered, only indexes on a primary key, or on unique or foreign keys can be proposed, and default values are assigned to the free space parameters.

The concurrency value specifies the amount of concurrent access to the table.

The frequency values are relative frequencies that specify how often an operation is used on the table.

Push Buttons

OK
Save
Reset
Cancel
Help

Table Partitions Window

Use this window to create new or modify existing ranges of the columns belonging to a partition:

For each column, specify the target UPPER value delimiting its range.

Push Buttons

- OK
- Save
- Reset
- Cancel
- Help

TeamConnection Database - Search Notebook

This notebook enables you to search for objects in the TeamConnection database.

Page

- General Page

Push Buttons

- Search
- Clear
- Cancel
- Help

General Page

This page enables you to do a narrow search or a broad search. A narrow search is a search for objects that belong to only one object type. For example, you are searching for the name of a DB2 system to fill in the **System** field of a **Populate** notebook. When you see this page, only 'DB2 System' appears in the **Object types** list -- the only relevant object type at the moment. You only have to decide whether to limit the search by specifying a full or partial object name. Then you select the **Search** push button.

A broad search is a search for objects that belong to many object types. It's the kind of search you typically perform from a workfolder. In this case, DataAtlas shows you a full list under **Object types** so that you can choose among them. Your choices are carried over to the **Types selected** list. As before, you next decide whether to limit the search by specifying a full or partial object name. The criteria you specify apply to all the object types you choose.

Lists

- Object types
- Types selected (for searches from a workfolder)

Fields

- Name
- Variation
- Revision

Name

This field contains an access name, a partial access name with a wildcard symbol, or a wildcard symbol alone. The field's contents are applied to all the selected object types during the search.

Do not specify other parts of a name (for example, relational qualifiers) in this field. You can specify variation and revision qualifiers in the **Variation** and **Revision** fields.

Required?: Yes. This field must contain a wildcard symbol at a minimum.

Variation

This field contains a variation qualifier, a partial variation qualifier with a wildcard symbol (%), or a wildcard symbol alone. The field's contents are applied to all the selected object types during the search.

Required?: No.

Revision

This field contains a revision qualifier, a partial revision qualifier with a wildcard symbol (%), or a wildcard symbol alone. The field's contents are applied to all the selected object types during the search.

Required?: No.

Object Types

The contents of this list depend on where the search request originated.

If the request originated from a workfolder...

The list identifies all the object types that can be stored in the TeamConnection database. You can select an object type to be included in the search by double clicking on it. This will move it to the **Types selected** list. To move many object types to the **Types selected** list:

1. Click them individually while holding down the Ctrl key or drag the mouse pointer through the list.

2. Click the **>** push button.

If the request originated elsewhere...

The list identifies only the object type that agrees with the source of the request. (For example, if you are searching for a DB2/390 table name, only "DB2/390 Table" appears.) The **Types selected** list is not shown.

Types Selected

This list is shown only if the search request originates from a workfolder. It contains the object types that you selected from the **Object types** list. These types are the ones DataAtlas searches for when you select the **Search** push button.

You can deselect an object type in this list by double-clicking on it. This will return it to the **Object types** list. To return many object types to the **Object types** list:

1. Click them individually while holding down the Ctrl key or drag the mouse pointer through the list.

2. Click the **&lt.** button.

Required?: Yes, at least one object type must be listed here to perform a search.

Search

Click this button to start the search for objects in the TeamConnection database that match your search criteria.

Matching objects are listed in the **TeamConnection - Search Results** window **.** When you select an object name from this window, the name is returned to the source of the search request.

Clear

Click this button to remove any search criteria that you typed in.

Wildcard Symbols in Names

To specify a partial access name or qualifier, use a percent sign (%) or an underscore (_) as wildcard symbols.   The underscore substitutes for only one character; the percent sign, for one or more characters.

You can use either wildcard symbol in front of, in the middle of, or at the end of a partial name, and you can use them in combination.

Examples

```
MyObject%
DB%Table
%Object
%test%
Var1_
_threv
%Object_
```

TeamConnection Database - Search Results Window

This window enables you to select an object from a list of candidates that belong to a specific object type.

List

Object List

Push Buttons

OK
Cancel
Help

Object List

This list displays the results of a search. The object or objects you choose from the list are returned to the place where the search request originated.

If you choose no object and select the **OK** push button, the first object in the list is selected by default.

Tip: To select all the objects in the list, press the Ctrl and / keys at the same time; to deselect all the selected objects, press the Ctrl and \ keys at the same time.

OK

Click this button to return the chosen object or objects to the place where the request for the search originated. Both this window and the **TeamConnection Database - Search** notebook close.

Cancel

Click this button to close this window without saving any selections you may have made. The **TeamConnection Database - Search** notebook remains open so that you can search again with different criteria if you choose.

Validation Report Window

In this window, you see the results of the performed actions for the validation of the current design of the selected objects.

The results are presented in terms of validation report items, which consist of an icon and a detailed description.

Each validation report item contains a warning about an invalid design step in your current design. On request, you get more information on a validation report item.

Also, you can look at the rule which caused a report item.

Tasks

- Evaluating the design support report

Push Buttons

- Cancel
- Help

Menu-Bar Choices

- Report
- Window
- Help

Pop-up menu

- Rule
- More

Validation Report Item Window

A report item reports the result of performing a previously selected action item.

You can request this window during the evaluation of a design support report.

In this window, you see the detailed description of a validation report item, and the objects affected by the performance of the action item.

Tasks

You can look at the notebook of an affected object by double-clicking on its icon.

Push Buttons

Cancel
Help

Workfolder Notebook

This notebook enables you to name the workfolder and assign TeamConnection classifications to it.

Pages

- General Page
- TeamConnection Page

Push Buttons

- OK
- Save
- Cancel
- Help

General Page

This page enables you to name the workfolder.   It also lets you specify a TeamConnection component and enter a description of the workfolder.

Fields

- Name
- Component
- Description
- History

Push Button

- Search

Name

This field contains the name of the workfolder.

Required?: No.

Default: The word 'Workfolder'.

Description

This field contains text that describes the workfolder.

Required?: No.

Limit: 4096 characters.

History

This read-only field shows, by date and time, when the workfolder was last modified.

TeamConnection Page

This page enables you to assign your workfolder to a family, release, and work area.

Fields

Family
Release
Work Area

Push Button

Search

Workfolder Window

In this window you can:

⇑ Search for data definitions in the TeamConnection database and put them in this workfolder

⇑ Create definitions directly in the TeamConnection database

⇑ Open the settings view of this workfolder and change its basic settings

Menu-Bar Choices

Workfolder
Selected
Edit
View
Window
Help

Workfolder

This choice displays a pull-down menu of actions you can perform with the workfolder you have opened.   The actions are:

Open as. Expands into these options...

⇑ **Icon view.** Shows objects as icons.

⇑ **Details view.** Shows each icon with its object name and object type.

⇑ **Flowed icon view.** Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column.   No up-and-down scrolling is needed to see all the icons.

Properties. Opens the notebook for the workfolder.   The notebook enables you to change the workfolder's name, description, and the TeamConnection classifications associated with it.

Save. Saves the contents of the workfolder without closing it.

Create object. Enables you to create all types of objects (DB2 tables, IMS DBDs, shareable data elements, and so on) in the TeamConnection database.

Creating an object does not put it into the workfolder.   To do that, you have to search for it in the TeamConnection database.   When DataAtlas finds your object, it adds the object to the workfolder.

Search TeamConnection database. Opens the **TeamConnection Database - Search** window, which enables you to search for objects in the TeamConnection database.

Close. Closes the workfolder. If it contains unsaved changes, DataAtlas asks you whether you want to save them.

Selected

This choice displays a pull-down menu of actions you can perform with objects you selected in this workfolder.   The actions are:

Properties. Opens the properties notebooks of the objects you selected.

Designer. This action is present only if the selected objects are DB2/390 objects. It expands into a menu of DataAtlas Designer functions that you can apply to the selected objects...

⇑ **Inform - Check completeness.**

⇑ **Validate - Check correctness.**

⇑ **Propose - Suggest improvements.**

Generate to file. Opens the **Generate to File** notebook, where you can export the definitions of selected objects to a file.

Copy. Copies the selected icons into another workfolder that has the same family, release, and work area.

Move. Moves the selected icons into another workfolder that has the same family, release, and work area.

Delete. Deletes the selected icons from the workfolder.   You can restore a deleted icon to the workfolder by searching the TeamConnection database for the object.

Original. Leads to a **Delete** option that enables you to delete objects in the TeamConnection database.   The icon for a deleted object is also deleted.

Edit

This choice displays two actions you can apply to the objects in this window:

Select all. Highlights (selects) all the objects in the window.

Deselect all. Removes highlighting from (deselects) all the objects in the window.

View

This choice controls the appearance and order of the objects in the workfolder. The items on its pull-down menu are:

Icon view. Shows objects as icons.

Details view. Shows each icon with its object name and object type.

Flowed icon view. Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Sort. Offers two choices, **Name** and **Type** . **Name** reorders the objects in the workfolder alphabetically by name. **Type** reorders them by object type. For example, all DB2 tables would be grouped together, as would all IMS DBDs.

Arrange. Arranges the icons to fit the size of a resized window.

Workfolders Folder

This folder contains your workfolders.   It also enables you to create new workfolders and to search for workfolders to put here.

Double-clicking on a workfolder in this folder opens it and shows the objects it contains.

Menu-Bar_Choices

Folder
Selected
Edit
View
Help

Folder

This choice displays three actions:

Open as. Expands into these options...

- ⇑ **Icon view.** Shows objects as icons.
- ⇑ **Flowed icon view.** Shows icons in a column and, at the bottom of the screen, flows additional icons to a new column. No up-and-down scrolling is needed to see all the icons.

Create workfolder. Creates a new workfolder. A workfolder is a "view" of the TeamConnection database in that it refers to a subset of the objects stored there.

Search for workfolders. Lets you search for workfolders in the TeamConnection database and place their icons in this window.

Selected

This choice displays a pull-down menu of actions that apply to the selected workfolder. The actions are:

Open as. Expands into these options...

⇑ **Icon view.** Shows objects as icons in the selected workfolder.

⇑ **Details view.** Shows icons with text in the selected workfolder.

⇑ **Flowed icon view.** Shows icons in a column in the selected workfolder.

Properties. Opens the properties notebook for the selected workfolder.

Delete. Deletes the icons of the selected workfolders.

Original. Leads to a **Delete** option that enables you to delete the selected workfolders from TeamConnection database.

Edit

This choice displays two actions you can apply to the profile icon.:

Select all. Highlights (selects) it.

Deselect all. Removes highlighting from (deselects) it.

Glossary

The definitions of terms used in the DataAtlas help information follow.

Access Name

The short, simple name of an object; the name by which it is best known outside the context of DataAtlas.

Alias

An alias is a locally defined name for a table or view in the same local or in a remote DB2 subsystem.   Aliases give DB2 location independence, because an alias can be created for a table at a remote site, thereby freeing the user from specifying the site that contains the data.

Aliases can also be used as a type of global synonym, because they can be accessed by anyone, not only by their creator.

Buffer Pool

You identify a buffer pool for several database objects. With the buffer pool you also determine the page size of the database object.   For 4K page buffer pools, the page size is 4K.   Otherwise, the page size is 32K.

Builder

An object that can transform one set of TeamConnection parts into another by invoking tools such as compilers and linkers.

Build Script

An executable or command file that specifies the steps that should occur during a TeamConnection build operation.

Build Tree

A graphical representation of the dependencies that the parts in a Team Connection application have on one another. If you change the relationship of one part to another, the build tree changes accordingly.

Database Descriptor (DBD)

A control block that describes an IMS/ESA database, including its organization and access method, the segments and fields in the database record, and the relationship between types of segments.

Data Element

The most elementary data type - for example, a field in an IMS segment or a column in a relational table.

DB2/390

An abbreviation for these DB2 enterprise data servers: DB2 Version 3, DB2 for MVS Version 4, and DB2 for OS/390 Version 5.

DB2 UDB

An abbreviation for DB2 Universal Database.

Generate

To export a data definition from the TeamConnection database that can be used in an operating environment.

Included Source Definition

An object that defines an included source file.   For COBOL, an included source file is a COPY file.

Local Data Component

A local data element or local data structure .

Local Data Element

The smallest unshared data unit within a data definition; for example, a column in a DB2 or Oracle table, a field in an IMS segment, or a COBOL or PL/I data item.

Local Data Structure

An unshared data structure containing local data elements .

Mapping Table

A table that contains a list of source object identifiers for local data components and a corresponding list of target object identifiers for shareable data components. The local data components are mapped to the shareable data components during populate processing. See also prototype mapping table.

Physical Design

An object that identifies the tables, indexes, table spaces, storage groups, and databases that you want to treat as a unit.

Populate

To import data definitions from external sources, such as an MVS PDS or a relational database catalog, into the TeamConnection database.

Prefix

The foremost qualifier in the names of some objects.   It identifies an owning object.   DataAtlas, not the user, adds it to the name, uses it, and maintains it.

Prototype Mapping Table

A mapping table DataAtlas produces when the mapping table being used has no entries for some of the local data components being processed. It is a prototype because the source names and target names in its entries are identical; you'll probably want to differentiate the names of your shareable data components.

Query

A request for information from the TeamConnection database based on specified conditions.

Reconcile

To define populated local data components in terms of shareable data components by using a mapping table.

Relational System

A name you use for grouping all the tables and associated objects that will be populated into the Team Connection database from a given DB2 UDB or DB2/390 catalog.

Relational Design

An object that identifies the tables you want to treat as a unit when using DataAtlas Modeler. (You must specify a relational design when you transform your tables from the relational definitions used by DB2 and Oracle to the entity-relationship model used by DataAtlas Modeler.)

Revision

An optional qualifier for the name of an object. It can be useful in distinguishing between similar objects that have to be available at the same time. For example, there could be two revisions of the same shareable data element named ZIPCODE: one representing a 9-digit zip code, one representing a 5-digit zip code, both co-existing within the same TeamConnection version. The revision qualifier could be used to differentiate them.

Segment

A basic unit of data in an IMS/ESA DBD.   The smallest amount of data that can be transferred by one DL/I operation or stored on a database.   A segment is made up of one or more fields.

Shareable Data Component

A shareable data element or shareable data structure.

Shareable Data Element

A TeamConnection object that can be used repeatedly to define data items like DB2 table columns, IMS database fields, and COBOL elementary items.

Shareable Data Structure

A TeamConnection object that can be used repeatedly to define data structures like IMS segments and COBOL group items.

Storage Group

A named set of DASD volumes on which DB2 data can be stored.

Supplied Query

One of a set of well-documented queries that is included with DataAtlas. The documentation includes a #NAME# and a #DESCRIPTION# section. These are loaded into the **Name** and **Description** fields on the **General** page of an **SQL Query** notebook when a supplied query is opened.

Synonym

A synonym is an alternative, private name for a table or a view. The synonym can only be used by the individual who created it.

Table

A named data object consisting of a specific number of columns and some number of unordered rows.

Table Space

In DB2, a page set used to store the records of one or more tables.

Trial Run

A hypothetical use of the Populate function.   DataAtlas does not import objects into the TeamConnection database but produces a populate report that shows what the result of an actual populate request would be.

Variation

An optional qualifier for the name of an object.   It can be useful in distinguishing between similar objects that have different uses.   For example, two shareable data elements could have the access name, ZIPCODE, but have different data characteristics for different applications.   The variation qualifier could designate the application to which each belongs.

Version

A given family, release, and work area within TeamConnection.

View

An alternative representation of data from one or more tables.   A view can include all or some of the columns contained in tables on which it is defined.

Workfolder

A place in which objects are collected that belong to the same family, release, and work area.

List of Error Messages

Select a message number in the list below to see the text of the message and an explanation.

- **EWS2001**
- **EWS2002**
- **EWS2003**
- **EWS2004**
- **EWS2005**
- **EWS2006**
- **EWS2007**
- **EWS2008**
- **EWS2031**
- **EWS2052**
- **EWS2055**
- **EWS2056**
- **EWS2057**
- **EWS2058**
- **EWS2059**
- **EWS2064**
- **EWS2065**
- **EWS2066**
- **EWS2073**
- **EWS2075**
- **EWS2089**
- **EWS2090**
- **EWS2091**
- **EWS2092**
- **EWS2093**
- **EWS2094**
- **EWS2095**
- **EWS2096**
- **EWS2097**
- **EWS2098**
- **EWS2099**
- **EWS2100**
- **EWS2101**
- **EWS2102**
- **EWS2103**
- **EWS2104**
- **EWS2105**
- **EWS2106**
- **EWS2107**
- **EWS2108**
- **EWS2109**
- **EWS2110**
- **EWS2111**
- **EWS2112**
- **EWS2113**
- **EWS2114**
- **EWS2115**
- **EWS2116**
- **EWS2117**
- **EWS2118**
- **EWS2119**
- **EWS2120**
- **EWS2121**
- **EWS2122**
- **EWS2123**
- **EWS2124**
- **EWS2125**
- **EWS2126**
- **EWS2127**
- **EWS2128**
- **EWS2129**
- **EWS2130**
- **EWS2131**
- **EWS2132**
- **EWS2133**
- **EWS2134**
- **EWS2135**
- **EWS2136**
- **EWS2137**
- **EWS2138**
- **EWS2139**
- **EWS2140**
- **EWS2141**
- **EWS2142**
- **EWS2143**
- **EWS2144**
- **EWS2145**
- **EWS2146**
- **EWS2147**
- **EWS2148**
- **EWS2149**
- **EWS2150**
- **EWS2151**
- **EWS2152**
- **EWS2153**
- **EWS2154**
- **EWS2158**
- **EWS2159**
- **EWS2160**
- **EWS2163**
- **EWS2164**
- **EWS2165**
- **EWS2166**
- **EWS2167**
- **EWS2168**
- **EWS2169**
- **EWS2170**
- **EWS2171**
- **EWS2172**
- **EWS2173**
- **EWS2174**
- **EWS2175**
- **EWS2176**
- **EWS2177**
- **EWS2178**
- **EWS2180**
- **EWS2181**
- **EWS2182**
- **EWS2183**
- **EWS2184**
- **EWS2185**
- **EWS2186**
- **EWS2187**
- **EWS2188**
- **EWS2189**
- **EWS2190**
- **EWS2191**
- **EWS2192**
- **EWS2195**
- **EWS2196**
- **EWS2197**
- **EWS2198**
- **EWS2199**
- **EWS2200**
- **EWS2201**
- **EWS2202**
- **EWS2203**
- **EWS2204**
- **EWS2205**
- **EWS2206**
- **EWS2212**
- **EWS2213**
- **EWS2214**
- **EWS2215**

EWS2216
EWS2217
EWS2218
EWS2219
EWS2220
EWS2221
EWS2222
EWS2223
EWS2224
EWS2225
EWS2226
EWS2227
EWS2228
EWS2229
EWS2230
EWS2250
EWS2251
EWS2252
EWS2253
EWS2254
EWS2255
EWS2256
EWS2257
EWS2258
EWS2259
EWS2260
EWS2261
EWS2262
EWS2263
EWS2269
EWS2270
EWS2271
EWS2272
EWS2273
EWS2274
EWS2275
EWS2276
EWS2277
EWS2278
EWS2279
EWS2280
EWS2289
EWS2291
EWS2292
EWS2293
EWS2295
EWS2296
EWS2297
EWS2298
EWS2299
EWS2300
EWS2301
EWS2302
EWS2303
EWS2304
EWS2305
EWS2311
EWS2313
EWS2315
EWS2338
EWS2339
EWS2340
EWS2343
EWS2344
EWS2353
EWS2355
EWS2356
EWS2357
EWS2358
EWS2375
EWS2376
EWS2377
EWS2378
EWS2390
EWS2391
EWS2392
EWS2393
EWS2394
EWS2395
EWS2397
EWS2398
EWS2401
EWS2402
EWS2403
EWS2404
EWS2405
EWS2406
EWS2407
EWS2409
EWS2410
EWS2413
EWS2414
EWS2415
EWS2416
EWS2417
EWS2418
EWS2419
EWS2420
EWS2422
EWS2423
EWS2425
EWS2427
EWS2428
EWS2429
EWS2430
EWS2431
EWS2432
EWS2433
EWS2434
EWS2435
EWS2436
EWS2437
EWS2438
EWS2439
EWS2440
EWS2441
EWS2442
EWS2443
EWS2444
EWS2449
EWS2461
EWS2462
EWS2463
EWS2464
EWS2465
EWS2466
EWS2467
EWS2468
EWS2469
EWS2470
EWS2471
EWS2472
EWS2474
EWS2475
EWS2476
EWS2479
EWS2481
EWS2483
EWS2484

EWS2485
EWS2486
EWS2487
EWS2488
EWS2489
EWS2490
EWS2491
EWS2497
EWS2510
EWS2512
EWS2513
EWS2517
EWS2518
EWS2519
EWS2520
EWS2526
EWS2527
EWS2529
EWS2530
EWS2531

EWS2001

Message: DataAtlas test message from < *where* >.

Explanation: A DataAtlas test message was invoked from *where* . This message is defined outside the TeamConnection messages file.

EWS2002

Message: DataAtlas cannot open the object *objectName* because it is incomplete. This object is probably referenced by some other object and was added to the TeamConnection database in an incomplete state when the other object was populated. You must populate this object before it can be opened.

Explanation: This error can occur after populating IMS objects or table objects. In the latter case, if the table has one or more foreign keys, its parent table may not have been populated. DataAtlas automatically creates objects to represent the table and shareable table definition of the parent. These objects are not complete until you explicitly populate the parent table. You cannot select or open a table or shareable table definition that is incomplete.

EWS2003

Message: The *objectType* named *objectName* already exists in the TeamConnection database. You must delete the existing object before you can populate it again.

Explanation: The fully qualified name of each object in the TeamConnection database must be unique among all the objects of the same type. If you wish to populate an object more than once, you must either delete the first instance before you populate the second instance or make sure that the fully qualified name of the second instance is different from that of the first. You can change the name by using a different variation or revision.

EWS2004

Message: Cannot allocate tcViewRoot.

Explanation: Call your IBM service representative for assistance.

EWS2005

Message: RDB method was invoked with no object to convert.

Explanation: Call your IBM service representative for assistance.

EWS2006

Message: RDB method was invoked with wrong attributes of the source object. The class name is < *className* >.

Explanation: DADB2PopGenTool method was passed wrong attributes from the source object. Check the table name, column definition, column type, unique column name, foreign column name, referential column name, base name, index name, view name, alias name, collection name, and so on.

EWS2007

Message: RDB method was invoked with the wrong class type.

Explanation: Refer to the log file for more details.

EWS2008

Message: You cannot add a unique column to the table < *objectName* >.

Explanation: If you want to add a unique column, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2031

Message: DataAtlas is unable to construct a shareable definition object because the name *name* is already in use.

Explanation: This problem may have occurred because you deleted a table or view that made use of the existing shareable definition but you did not delete the definition. In this case, you should delete the shareable definition now. Otherwise, you may need to chose a different name, variation, or revision for the table or view that you want to create.

EWS2052

Message: The object you are trying to open has been updated in the TeamConnection database since you last opened it. If you would like to refresh your copy with the current copy from the database, click 'Yes'. If you would like to cancel the open request, click 'No'.

Explanation: You cannot open an object if it has been changed in the TeamConnection database, because any changes you might make cannot be stored. If you choose to refresh your copy, any uncommitted changes you may have made to this object elsewhere will be overwritten. If you choose not to refresh your copy, you cannot make any changes to the object until you refresh it later, or until you shut down DataAtlas and restart it.

EWS2055

Message: The length specified for a column is not a positive integer.

Explanation: Change the length field to a positive integer and try again.

EWS2056

Message: The length exceeds the maximum of *maxVal* characters for data type *type* .

Explanation: Make the length field less than or equal to the allowed maximum and try again.

EWS2057

Message: The scale exceeds the precision of the decimal number.

Explanation: Modify the scale or precision field and try again.

EWS2058

Message: Query < *queryName* > cannot be opened because the file in which it was stored cannot be read.

Explanation: Query < *queryName* > was previously saved into the file < *queryFile* >. That file cannot be read. It may no longer exist in the file system or it may be damaged. If you no longer need this SQL query, you can go back to the folder and delete it.

EWS2059

Message: Report < *reportName* > cannot be opened because the file in which it was stored cannot be read.

Explanation: Report < *reportName* > was previously saved into the file < *reportFile* >. That file cannot be read. It may no longer exist in the file system or it may be damaged. If you no longer need this report, you can go back to the folder and delete it.

EWS2064

Message: Query < *queryName* > cannot be written to file < *fileName* >.

Explanation: Ensure that the file name is correct. If the file already exists, ensure that it is not read-only or already in use.

EWS2065

Message: Report < *reportName* > cannot be written to file < *fileName* >.

Explanation: Ensure that the file name is correct. If the file already exists, ensure that it is not read-only or already in use.

EWS2066

Message: You must specify an SQL statement before < *queryName* > can be run.

Explanation: Turn to the Query page of the notebook and specify a valid SELECT statement.

EWS2073

Message: The file < *fileName* > cannot be loaded.

Explanation: The file must be a valid query file. Check that the file exists and that it is not currently being used by another application, and retry the action.

EWS2075

Message: The report could not be exported to file < *fileName* >.

Explanation: Ensure that the file name is correct. If the file already exists, ensure that it is not read-only or already in use.

EWS2089

Message: A name does not conform to IMS naming conventions.

Explanation: IMS naming conventions require names to begin with a nonnumeric character and consist only of the characters 0-9, A-Z, @, #, and $. A DBD name, DDname, or routine name does not meet this requirement.

EWS2090

Message: The value range for Root Anchor points is 1 to 255.

Explanation: The value entered for Root Anchor Points is outside the valid range.

EWS2091

Message: The value range for Relative Block Number is 1 to 16777215.

Explanation: The value for Relative Block Number is outside the valid range.

EWS2092

Message: The value range for Bytes 1 to 16777215.

Explanation: The value entered for randomizing module bytes is outside the valid range.

EWS2093

Message: The DDname must conform to IMS naming conventions.

Explanation: IMS naming conventions require the DDname to begin with a nonnumeric character and consist only of the characters 0-9, A-Z, @, #, and $.

EWS2094

Message: The Label must conform to IMS naming conventions.

Explanation: IMS naming conventions require the Label to begin with a nonnumeric character and consist only of the characters 0-9, A-Z, @, #, and $.

EWS2095

Message: The value range for Scan Cylinders is 0 to 255.

Explanation: The value entered for Scan Cylinders is outside the valid range.

EWS2096

Message: The value range for Size is 1 to 32767.

Explanation: The value entered for Size is outside the valid range.

EWS2097

Message: The calculated block size is greater than 32760.

Explanation: The record length used for the calculation has 2 added if the operating system access method is BSAM. The final block size has 4 added to it if the record format is variable.

EWS2098

Message: The value range for Blocking Factor is 1 to 32767.

Explanation: The value entered for Blocking Factor is outside the valid range.

EWS2099

Message: The value range for Free Block Frequency Factor is 0 to 100, except 1.

Explanation: The value entered for Free Block Frequency Factor is outside the valid range.

EWS2100

Message: The value range for Free Space Percentage Factor is 1 to 99.

Explanation: The value entered for Free Space Percentage Factor is outside the valid range.

EWS2101

Message: The value range for Units of Work and Root is 2 to 32767.

Explanation: The value entered for Units of Work or Root is outside the valid range.

EWS2102

Message: Units of work and Overflow must be specified together, with the value for Overflow smaller than the value for Units of Work.

Explanation: Units of work or Overflow was specified alone or the Overflow value is larger than the Units of work value.

EWS2103

Message: Root and Overflow must be specified together, with the value for Overflow smaller than the value for Root.

Explanation: The Root or Overflow was specified alone or the Overflow value is larger than the Root value.

EWS2104

Message: The value range for Frequency is 0.01 to 16777215.

Explanation: The value for Frequency is outside the valid range.

EWS2105

Message: The value range for Segment Bytes is 1 to *bytes* .

Explanation: The value for Bytes, Maximum or Minimum, is outside the valid range.

EWS2106

Message: Maximum bytes must be larger than Minimum bytes.

Explanation: The value for Minimum is greater than the value for Maximum.

EWS2107

Message: The value range for Subset pointers is 1 to 8.

Explanation: The value for Subset pointers is outside the valid range.

EWS2108

Message: The value for Size must be a multiple of 512 for values less than 8192, and a multiple of 2048 for values up to 30720.

Explanation: The value for Size is not valid for a data set in a DBD that uses VSAM.

EWS2109

Message: The value for Size must be a multiple of 512 for values less than 8192, and a multiple of 4096 for values up to 28672.

Explanation: The value for Size is not valid for a DEDB DBD.

EWS2110

Message: PCB name must not begin with DFS.

Explanation: The value entered for the PCB name begins with a reserved prefix.

EWS2111

Message: The Processing options value is not valid.

Explanation: The value entered for Processing options is not valid.

EWS2112

Message: The value range for Key length is 1 to 32767.

Explanation: The value entered for Key length is outside the valid range.

EWS2113

Message: The value range for Maximum Qx calls is 0 to 32767.

Explanation: The value entered for Maximum Qx calls is outside the valid range.

EWS2114

Message: The value range for I/O area size is 0 to 256000.

Explanation: The value entered for I/O area size is outside the valid range.

EWS2115

Message: The value range for SSA size is 0 to 256000.

Explanation: The value entered for SSA size is outside the valid range.

EWS2116

Message: The value range for I/O error option is 0 to 4095.

Explanation: The value entered for I/O error option is outside the valid range.

EWS2117

Message: I/O error option must be specified when Write to operator is specified.

Explanation: Write to Operator is specified, but no I/O Error option was given.

EWS2118

Message: ASSEM must be specified for Language when Online Image Copy is specified.

Explanation: Online Image Copy was specified, but the language is not ASSEM.

EWS2119

Message: Calculated block size of *blocksize* exceeds limit of 32,760.

Explanation: The product of the blocking factor and the record length exceeds the limit.

EWS2120

Message: The value range for Record Length is 1 to 30713.

Explanation: The record length is outside the valid range.

EWS2121

Message: Blocking Factor and Size cannot both be specified.

Explanation: The Blocking Factor and Size both have values greater than 0.

EWS2122

Message: DDname *ddname* is a reserved word.

Explanation: The string entered for the DDname is a name reserved for system use.

EWS2123

Message: This DBD cannot have another data set.

Explanation: A HDAM or HIDAM DBD can have only 10 data sets.

EWS2124

Message: This segment cannot have another field.

Explanation: A segment can have only 255 fields.

EWS2125

Message: This segment cannot have another secondary index.

Explanation: A segment can have only 32 secondary indexes.

EWS2126

Message: This DBD cannot have another segment.

Explanation: There can be only 255 segments in an HDAM, HIDAM, HISAM, HSAM, or LOGICAL DBD, only 127 segments in a DEDB DBD, or only 1 segment in an INDEX, MSDB, SHSAM, or SHISAM DBD.

EWS2127

Message: This secondary index cannot add another search field.

Explanation: A secondary index can have only 5 fields defined as search fields.

EWS2128

Message: This secondary index cannot add another subsequence field.

Explanation: A secondary index can have only 5 fields defined as subsequence fields.

EWS2129

Message: This secondary index cannot add another duplicate data field.

Explanation: A secondary index can have only 5 fields defined as duplicate data fields.

EWS2130

Message: The value range for Blocking Factor is 1 to 30709.

Explanation: A data set in an HDAM or HIDAM DBD that uses VSAM specifies a blocking factor value outside the valid range for size without overhead.

EWS2131

Message: Maximum record length is less than minimum record length.

Explanation: The data set in a GSAM DBD specifies a smaller value for the maximum record length than for the minimum record length.

EWS2132

Message: Maximum record length must equal minimum for fixed record format.

Explanation: The data set in a GSAM DBD specifies a fixed-length record format, but has a value for minimum record length that is neither 0 nor the same as the maximum record length.

EWS2133

Message: Input record length must not be less than output record length.

Explanation: The data set specifies an output record length longer than the input record length.

EWS2134

Message: The length in this segment must be divisible by 4.

Explanation: The value for Bytes is not valid.

EWS2135

Message: Compaction routine name *ddname* is a reserved word.

Explanation: The string entered for the compaction routine name reserved for system use.

EWS2136

Message: The value of Type conflicts with value of Bytes for this field.

Explanation: The value of Type is F (fullword) and Bytes is not 4, or the value of Type is H (halfword) and the Bytes is not 2.

EWS2137

Message: The value range for maximum locks is 0 to 255.

Explanation: The maximum number of locks specified is outside the valid range.

EWS2138

Message: Index fields on this secondary index have a combined length greater than 240.

Explanation: The search, subsequence and duplicate data fields have a combined length greater than 240 bytes.

EWS2139

Message: Field *name* is not valid as a Duplicate Data field.

Explanation: A system-related field was selected as a duplicate data field.

EWS2140

Message: Teleprocessing PCB *name* must precede all database PCBs and GSAM PCBs.

Explanation: A teleprocessing PCB was found after database or GSAM PCBs.

EWS2141

Message: Database PCB *name* must follow all teleprocessing PCBs and precede all GSAM PCBs.

Explanation: A database PCB was found before teleprocessing PCBs or after GSAM PCBs.

EWS2142

Message: GSAM PCB *name* must follow all teleprocessing PCBs and database PCBs.

Explanation: A GSAM PCB was found before teleprocessing or database PCBs.

EWS2143

Message: Processing option *name* is not valid when a sequence DBD is specified for this PCB.

Explanation: Processing options of L and LS are invalid with on a PCB using a secondary processing sequence.

EWS2144

Message: Base DBD *name* has access method *access*

Explanation: A GSAM PCB must be based on a GSAM DBD.

EWS2145

Message: This sensitive segment cannot have another sensitive field.

Explanation: A sensitive segment can have only 255 sensitive fields.

EWS2146

Message: This PSB cannot have another PCB.

Explanation: A PSB can contain only 459 PCBs.

EWS2147

Message: This DBD cannot have another area.

Explanation: A DEDB DBD can contain only 240 areas.

EWS2148

Message: Base DBD *name* has access method *access* .

Explanation: A database PCB cannot be based on a GSAM DBD or on an INDEX DBD.

EWS2149

Message: DD2 cannot be the same as DD1.

Explanation: The string entered for the second DDname is not unique.

EWS2150

Message: This segment cannot have another logical child relationship.

Explanation: A segment can have only 255 logical children.

EWS2151

Message: The value range for Minimum Bytes on a DEDB segment is 4 to 32760.

Explanation: The value for Minimum Bytes is outside the valid range for a DEDB segment.

EWS2152

Message: The value range for Bytes on a field is 1 to 256.

Explanation: The value for Bytes is outside the valid range for a field. If the field name starts with '/CK', the maximum field length is 3825.

EWS2153

Message: The value range for start position on a field is 1 to 32767.

Explanation: The value for the starting position is outside the valid range for a field. If the field name starts with '/CK', the maximum field start position is 3825.

EWS2154

Message: The file < *helpFile* > could not be found. No help information will be available within DataAtlas.

Explanation: The file < *helpFile* > contains DataAtlas help information, but the file could not be found. To correct the problem, ensure that the file exists in one of the directories identified by the 'HELP' environment variable, and restart DataAtlas.

EWS2158

Message: There is an invalid value in the *fieldName* field.

Explanation: Valid values are described in the online help documentation. Correct the data so that the window can be accepted. Available actions may include: 1) OK - return to the panel to correct the data, 2) Cancel - ignore changes made since the last store or retrieve and continue processing, 3) Copy to clipboard - copy this message to the clipboard, 4) Help - display detailed help for this message.

EWS2159

Message: A value must be specified in the *fieldName* field.

Explanation: Valid values are described in the online help documentation. Correct the data so that the window can be accepted. Available actions may include: 1) OK - return to the panel to correct the data, 2) Cancel - ignore changes made since the last store or retrieve and continue processing, 3) Copy to clipboard - copy this message to the clipboard, 4) Help - display detailed help for this message.

EWS2160

Message: This notebook has a field with invalid data. Please make sure mandatory fields are not blank.

Explanation: The notebook has a blank mandatory field or a field with invalid data.

EWS2163

Message: An attempt to recreate a named object was made.

Explanation: *tcError*

EWS2164

Message: The search for an object yielded no matches.

Explanation: *tcError*

EWS2165

Message: A reference was made to a object type that is not a supported by TeamConnection.

Explanation: *tcError*

EWS2166

Message: The store operation failed because of constraints violation.

Explanation: *tcError*

EWS2167

Message: The version string is invalid.

Explanation: *tcError*

EWS2168

Message: The specified object cannot be copied.

Explanation: *tcError*

EWS2169

Message: The specified object cannot be deleted.

Explanation: *tcError*

EWS2170

Message: The specified part was not found.

Explanation: *tcError*

EWS2171

Message: The level of the Data Item being generated exceeds the language limit.

Explanation: *tcError*

EWS2172

Message: A processing error occurred in Team Connection.

Explanation: *tcError*

EWS2173

Message: A processing error occurred in Team Connection Cache Layer.

Explanation: *tcError*

EWS2174

Message: An attempt to access a file failed.

Explanation: *tcError*

EWS2175

Message: An attempt to store an object failed.

Explanation: *tcError*

EWS2176

Message: Team Connection has returned with an unexpected error.

Explanation: *tcError*

EWS2177

Message: The Alias or Synonym < *objectName* > already exists in TeamConnection database. You must delete the existing object before you can populate it again.

Explanation: The fully qualified name of each object in the TeamConnection database must be unique among all the objects of the same type. If you wish to populate an object more than once, you must either delete the first instance before you populate the second instance or make sure that the fully qualified name of the second instance is different from that of the first. You can change the name by using a different DB2 system, variation, or revision.

EWS2178

Message: You cannot alter the ROSHARE attribute of the database *objectName* .

Explanation: You cannot alter the database *objectName* because the ROSHARE attribute of the DB2 database in either the TeamConnection database or in the DB2 catalog has the value READ. The ROSHARE attribute can be altered only if the new and old values are either OWNER or NONE. You must use DROP and CREATE to modify this database in the DB2 catalog. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2180

Message: You cannot alter the number of partitions used by the table space < *objectName* >.

Explanation: If you want to change the number of partitions, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2181

Message: You cannot alter the segment size used by the table space < *objectName* >.

Explanation: If you want to change the segment size, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2182

Message: You cannot alter the UNIQUE attribute of the index < *objectName* >.

Explanation: If you want to change the UNIQUE attribute of the index, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2183

Message: You cannot alter the table referenced by the index < *objectName* >.

Explanation: If you want to change the table, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2184

Message: The index < *objectName* > cannot be altered because you cannot change the cluster index of a table using an ALTER statement.

Explanation: If you want to change the cluster index, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2185

Message: You cannot alter the number of partitions used by the index < *objectName* >.

Explanation: If you want to change the number of partitions, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2186

Message: You cannot alter the VALUES of any of the partitions used by the index < *objectName* >.

Explanation: If you want to change the VALUES, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2187

Message: You cannot alter the SUBPAGES attribute of the index < *objectName* >.

Explanation: If you want to change the SUBPAGES, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2188

Message: You cannot add, remove, or modify any of the columns of the index < *objectName* > using an ALTER statement.

Explanation: If you want to change the index columns, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2189

Message: You cannot alter the DEFER attribute of the index < *objectName* >.

Explanation: If you want to change the DEFER attribute, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2190

Message: You cannot alter the catalog used by the storage group < *objectName* >.

Explanation: If you want to change the catalog, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2191

Message: DataAtlas cannot generate the object *objectName* because it is incomplete. This object is probably referenced by some other object and was added to the TeamConnection database in an incomplete state when the other object was populated. You must populate this object before it can be generated.

Explanation: This error can result from an incomplete IMS object or table object. In the latter case, if the table has one or more foreign keys, its parent table may not have been populated. DataAtlas automatically creates objects to represent the parent table. These objects are not complete until you explicitly populate the parent table. You cannot generate an incomplete table.

EWS2192

Message: The DDL generated for the object *objectName* contains columns with undefined data types. Edit this DDL to make sure valid values are specified for the data types of all columns.

Explanation: Depending on the version of DB2 that you are using, there may be some DB2 data types that are not supported by the current version of DataAtlas. If there are any columns that use unsupported data types, DataAtlas generates the data type as 'UNDEFINED'. You have to modify the generated DDL before it can be submitted to DB2.

EWS2195

Message: Cannot connect to the database *databaseName* . The SQLCODE received from the database is *sqlCode* .

Explanation: *DB2 error message*

EWS2196

Message: The DB2/CS object cannot be created because the specified system *systemName* is not a DB2/CS system.

Explanation: Choose a system of the correct type and try again.

EWS2197

Message: The DB2/390 object cannot be created because the specified system *systemName* is not a DB2/390 system.

Explanation: Choose a system of the correct type and try again.

EWS2198

Message: An internal error occurred while processing SQL statements.

Explanation: Incorrect SQL statements have been generated, causing an internal processing error. Call your IBM service representative for assistance.

EWS2199

Message: You cannot alter the unique key of the table < *objectName* >.

Explanation: If you want to make changes to a unique key, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2200

Message: You cannot alter the table space used by the table < *objectName* >.

Explanation: If you want to change the table space, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2201

Message: You cannot alter the database used by the table < *objectName* >.

Explanation: If you want to change the database, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2202

Message: You cannot alter the OBID used by the table < *objectName* >.

Explanation: If you want to change the OBID, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2203

Message: You cannot alter the EDITPROC used by the table < *objectName* >.

Explanation: If you want to change the EDITPROC, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2204

Message: You cannot alter the columns of the table < *objectName* >.

Explanation: The only change you can make to the columns of a table with an ALTER statement is to add columns at the end of the list of columns. If you want to make any other changes to the columns of a table, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2205

Message: The object type being populated is an IMS PSB, but the source file contains IMS DBD macro statements.

Explanation: The first macro statement in the source file is DBD, rather than the expected PCB or PSBGEN statement. Make a new populate request, specifying the correct object type.

EWS2206

Message: DBD <> *dbdName* < *variation* : *revision* > must be populated before the current PCB or logical DBD.

Explanation: The current PCB or logical DBD cannot be populated because the base or source DBD has not been populated. Populate physical DBDs before the logical DBDs that depend on them. Populate all DBDs before PSBs.

EWS2212

Message: Cannot bind the DataAtlas application program to database *databaseName* . The SQLCODE received from the database is *sqlCode* .

Explanation: Refer to the DB2 manuals for more information on the SQLCODE.

EWS2213

Message: An error occurred while trying to obtain a list of database aliases. The returned SQLCODE is *sqlCode* .

Explanation: This may happen because DB2 has not been started or because there was a problem with one of the database aliases. Refer to the DB2 manuals for more information on the returned SQLCODE, and take any indicated action to solve the problem. If the problem still exists, call your IBM service representative for assistance.

EWS2214

Message: A DataAtlas internal error has occurred.

Explanation: Call your IBM service representative for assistance.

EWS2215

Message: DataAtlas cannot list tables *objectName* from the selected database. The returned SQLCODE is *sqlCode* .

Explanation: This may be due to a database connection problem, an error in the DB2 database catalog tables, or a DataAtlas internal error. Refer to the DB2 manuals for more information on the returned SQLCODE and take any indicated action to solve the problem. If the problem still exists, call your IBM service representative for assistance.

EWS2216

Message: DataAtlas cannot extract table *objectName* from the selected database. The returned SQLCODE is *sqlCode* .

Explanation: This may be due to a database connection problem, an error in the DB2 database catalog tables, or a DataAtlas internal error. Refer to the DB2 manuals for more information on the returned SQLCODE, and take any indicated action to solve the problem. If the problem still exists, call your IBM service representative for assistance.

EWS2217

Message: DataAtlas cannot extract table *objectName* from the selected database.

Explanation: This may be due to an error in the DB2 database catalog tables or a DataAtlas internal error. Call your IBM service representative for assistance.

EWS2218

Message: DataAtlas was unable to locate the object *objectName* in the selected database. No Alter DDL has been generated.

Explanation: Verify that the correct database alias was selected. If not, select the correct database alias and try again. Otherwise, it is possible that the object has been dropped from the database. If this is the case and you want to recreate it, click the Create button instead of the Alter button in the Generate notebook.

EWS2219

Message: You specified one or more invalid characters in the compiler options for a COBOL populate or generate.

Explanation: Valid options are listed in the documentation for the IBM VisualAge for COBOL for OS/2 compiler. Examples of invalid characters in compiler options are the double-quote (") and minus sign (-). Correct the options specified on the Compiler page of the COBOL notebook or in your profile.

EWS2220

Message: You cannot populate < *objectName* > because it already exists in the cache or the TeamConnection database. You must delete or rename the existing object before you can populate it again.

Explanation: The fully qualified name of each object in the TeamConnection database must be unique among all the objects of the same type. If you wish to populate an object more than once, you must either delete the first instance before you populate the second instance or make sure that the fully qualified name of the second instance is different from that of the first. You can change the name by using a different DB2 system, variation, or revision.

EWS2221

Message: The file, *fileName* , cannot be accessed.

Explanation: It may not exist, or it may be being accessed by another process.

EWS2222

Message: AN OPERATING SYSTEM CALL, *SYSTEMCALL* , FAILED. The return code is: *returncode*

Explanation: Call your IBM service representative for assistance.

EWS2223

Message: The *compilertype* compiler returned with a non-zero return code= *returncode* . A compilation error or other problem was detected.

Explanation: Consult the compiler documentation for the meaning of the return code. If you still need assistance, call your IBM service representative.

EWS2224

Message: An unrecognized exception has occurred. < *code* = *value* >. The condition code for the exception code cannot be found in the TeamConnection Message Table. See the TeamConnection logs for details as to where the exception occurred.

Explanation: The condition code for the exception code cannot be found in the TeamConnection message table. See the TeamConnection logs for details as to where the exception occurred.

EWS2225

Message: An internal processing error occurred while attempting to add an object, *objectname* to a collection.

Explanation: Call your IBM service representative for assistance.

EWS2226

Message: An unexpected attempt was made to call a method that is not implemented in the DSObject subclass against which it was invoked. The object type against which the method was invoked is *methodname1* . The method that is invoked was *methodname2* .

Explanation: Call your IBM service representative for assistance.

EWS2227

Message: The source file contains an unexpected macro statement or keyword.

Explanation: Either there is an invalid keyword or parameter in the source file, there is unexpected punctuation, or the order of statements is invalid. Verify that the source file can be run through DBDGEN or PSBGEN.

EWS2228

Message: The object type being populated is an IMS DBD, but the source file contains IMS PSB macro statements.

Explanation: The first macro statement in the source file is not DBD, but is a PCB or PSBGEN statement. Make a new populate request, specifying the correct object type.

EWS2229

Message: You have executed a method that requires the DataAtlas resource library *resourceLib* . That resource library cannot be found.

Explanation: The DataAtlas resource library *resourceLib* cannot be found. You may be invoking a DataAtlas method incorrectly specifying a view type that is not known by DataAtlas.

EWS2230

Message: The TC family could not be contacted.

Explanation: *tcError*

EWS2250

Message: ERROR:

EWS2251

Message: ERROR: Failure in populating program source into work area.

EWS2252

Message: ERROR: Failure in analyzing the program source.

EWS2253

Message: ERROR: TSO-E command processor encountered an error. CC= *var1*

EWS2254

Message: ERROR: TSO-E command processor encountered a severe error.

EWS2255

Message: ERROR: TSO-E command processor encountered an unrecoverable error.

EWS2256

Message: ERROR: An error code was returned from compilation of < *var1* >.

EWS2257

Message: ERROR: A severe error code was returned from compilation of < *var1* >.

EWS2258

Message: ERROR: An unrecoverable error code was returned from compilation of < *var1* >.

EWS2259

Message: ERROR: File < *var1* > could not be opened for WRITE.

EWS2260

Message: ERROR: File < *var1* > could not be opened for READ.

EWS2261

Message: ERROR: An unexpected error code was returned from compilation of < *var1* >.

EWS2262

Message: ERROR: Attempt to set name < *var1* > failed. Name already exists.

EWS2263

Message: ERROR: Invalid character detected in options < *var1* > specified for COBOL populate/generate.

EWS2269

Message: ERROR:

EWS2270

Message: ERROR: Failure in populating program source into work area.

EWS2271

Message: ERROR: Failure in analyzing the program source.

EWS2272

Message: ERROR: TSO-E command processor encountered an error. CC= *var1* .

EWS2273

Message: ERROR: TSO-E command processor encountered a severe error.

EWS2274

Message: ERROR: TSO-E command processor encountered an unrecoverable error.

EWS2275

Message: ERROR: An error code was returned from compilation of < *var1* >.

EWS2276

Message: ERROR: A severe error code was returned from compilation of < *var1* >.

EWS2277

Message: ERROR: An unrecoverable error code was returned from compilation of < *var1* >.

EWS2278

Message: ERROR: File < *var1* > could not be opened for WRITE.

EWS2279

Message: ERROR: File < *var1* > could not be opened for READ.

EWS2280

Message: ERROR: An unexpected error code was returned from compilation of < *var1* >.

EWS2289

Message: This notebook has a field with invalid data. Please make sure mandatory fields are not blank.

Explanation: The notebook has a blank mandatory field or a field with invalid data.

EWS2291

Message: Required field not completed.

EWS2292

Message: The Name field must not be empty. Please fill in a value and try again.

EWS2293

Message: The System field must not be empty. Please search for an object and try again.

EWS2295

Message: The Creator field must not be empty. Please search for an object and try again.

EWS2296

Message: A object with the same name already exists. Please change the name and try again.

EWS2297

Message: No connection to a DB2/CS server could be established. Please try again later or ask your local administrator.

EWS2298

Message: A problem occurred during the extraction of DB2/390 statistic data. Please try again later or ask your local administrator.

EWS2299

Message: The Database field must not be empty. Please search for an object and try again.

EWS2300

Message: The corresponding table partitions do not match with the index partitions. The index partitions will be automatically updated when you open the index notebook. These changes will be made in the TeamConnection database when you save the index.

EWS2301

Message: The Primary Quantity value exceeds the maximum of 4194304.

EWS2302

Message: The Secondary Quantity value exceeds the maximum of 131068.

EWS2303

Message: The Table field on the Definition page must not be empty. Please search for an object and try again.

EWS2304

Message: Failed to delete all rows

Explanation: Exception

EWS2305

Message: Failed to delete last row

Explanation: Exception

EWS2311

Message: *errMsg*

Explanation: Query Failed

EWS2313

Message: You cannot change the primary key. There are foreign keys assigned to the primary key.

EWS2315

Message: New Table *name variation revision q3 q4* cannot be created

EWS2338

Message: The operation *var1* could not be executed because the tool < *var2* > against which that operation must be invoked could not be found.

Explanation: An attempt was made to execute an operation *var1* against a tool *var2* which resides in a DLL *var3* which must be dynamically loaded, but either the tool or the DLL could not be found. Please verify that the DLL is in your libpath and try the operation again.

EWS2339

Message: The specified input parameter *var1* is invalid.

Explanation: The input parameter specified was found to be invalid. Please contact your system administrator.

EWS2340

Message: The DLL *var1* could not be dynamically loaded. The error that occurred was *var2* .

Explanation: An attempt was made to dynamically load the DLL *var1* , but either the DLL could not be found or it cannot be loaded for this level of the product. Please verify that the DLL is in your libpath and that it can be loaded and try the operation again.

EWS2343

Message: The Column Name field must not be empty and must be a valid SQL identifier. Fill in a value and try again.

EWS2344

Message: The type of the relational system is incorrect. Specify a valid DB2/CS or DB2/390 type.

EWS2353

Message: Command file does not exist.

Explanation: The file specified on the Command file parameter does not exist.

EWS2355

Message: *errMsg*

EWS2356

Message: No databases returned

EWS2357

Message: There must be at least one object type added to the right list box.

Explanation: No object type selected

EWS2358

Message: *errMsg*

Explanation: Foreigh Key Save Error

EWS2375

Message: Reconcile flag selected without specifying a mapping table.

EWS2376

Message: Mapping table does not exist.

Explanation: The mapping table specified does not exist.

EWS2377

Message: The following required fields are empty: *field*

EWS2378

Message: You must specify values for all the fields in this window in order to start DataAtlas.

These become your default values for the TeamConnection family, release, workarea, and component. Once you start using DataAtlas, if you later decide you want to use a different value for one of these fields, you can indicate changes on the TeamConnection page of the DataAtlas Profile.

Explanation: DataAtlas needs initial values for the TeamConnection family, release, workarea, and component you will use. After you've supplied values for these fields and the DataAtlas Main Window appears, you can use the DataAtlas Profile to switch to a different family, release, workarea, or component whenever you'd like.

EWS2390

Message: File name must be entered to save the query

EWS2391

Message: The file is a read-only file and cannot be overwritten. Please specify a different name for the file in which to save the query.

EWS2392

Message: No valid copy objects selected.

EWS2393

Message: No valid delete objects selected.

EWS2394

Message: The following object(s) are still being created. If you choose to continue, they will NOT be stored in the workfolder. After you finish creating the object(s) you can search for them from the workfolder to add them back into the workfolder.

objectList

EWS2395

Message: The workfolder *name* could not be stored. You can either discard your workfolder changes or return to the workfolder to try again.

Explanation: An attempt was made to store changes to workfolder *name* but the store attempt failed. You must decide whether to exit out of this workfolder without saving changes by choosing the Discard Changes button, or return to the opened workfolder and attempt the operation again by selecting Return to Workfolder.

EWS2397

Message: This workfolder no longer exists in the TeamConnection database. Would you like to proceed to create a new one?

EWS2398

Message: A blank name field is not allowed.

EWS2401

Message: The parameter value <%1> is invalid. Valid values range from <%2> to <%3>.

EWS2402

Message: The parameter value for SUBPAGES <%1> is invalid. Valid values for SUBPAGES are: 1, 2, 4, 8, 16.

EWS2403

Message: The value of parameter 2 <%1> is less than or equal to parameter 1 <%2>.

Explanation: The value of parameter 1 must be less than the value for parameter 2.

EWS2404

Message: The default %1 value for %2 pages <%3> is invalid. Valid values range from <%4> to <%5>.

EWS2405

Message: The default SEGSIZE value <%1> is invalid. Valid values for the SEGSIZE are multiples of 4, between 4 and 64.

EWS2406

Message: The value of parameter 1 <%1> is invalid. Valid values are: 0,1,2,...,49.

EWS2407

Message: The value of parameter 2 <%1> is invalid. Valid values are: 0,1,2,3,..,9.

EWS2409

Message: A check constraint with the same name already exists. Please change the name and try again.

EWS2410

Message: RDB Platform *ePlatForm* unknown or not supported.

EWS2413

Message: AN OPERATING SYSTEM CALL, *SYSTEMCALL* , FAILED. The return code is: *returncode*

Explanation: Call your IBM service representative for assistance.

EWS2414

Message: An attempt was made to invoke the *compilertype* compiler using an operating system call. A non-zero return code= *returncode* was received indicating that the compiler could not be found. If you have checked the "Use command file note compiler options" box, it is also possible the command file was not found.

Explanation: Make sure your compiler is installed correctly and can be invoked from the command line in the session in which you plan to run DataAtlas. If you wish to use a command file, make sure it exists. If the problem persists, call your IBM service representative for assistance.

EWS2415

Message: While processing a non-zero return code= *returncode* from the *compilertype* compiler, a problem was encountered with the listing file, *filename* . The listing file could not be opened or could not be closed.

Explanation: Make sure the listing file is not in use. If the problem persists, Call your IBM service representative for assistance.

EWS2416

Message: You cannot alter the WHERE NOT NULL attribute of the UNIQUE Index < *objectName* >.

Explanation: If you want to change the WHERE NOT NULL attribute of the unique index, you must generate DROP and CREATE statements. If you execute these statements, any data that is currently in your DB2 database will be lost.

EWS2417

Message: You cannot alter the Convert from a Type 2 to a Type 1 Index < *objectName* >.

Explanation: You cannot alter the Convert from a Type 2 to a Type 1 Index. The database might be out of synch.

EWS2418

Message: You cannot alter the Locksize Row from table space < *objectName* >.

Explanation: You cannot alter the Locksize Row from table space. The database might be out of synch.

EWS2419

Message: You cannot alter the CCSID < *objectName* >.

Explanation: You cannot alter the CCSID. The database might be out of synch.

EWS2420

Message: You cannot alter the LARGE attribute of the table space < *objectName* >.

Explanation: You cannot alter the LARGE attribute of the table space. The database might be out of synch.

EWS2422

Message: You need to select a table space of type LONG for "Long In".

Explanation: The "Long In" table space should be of type LONG.

EWS2423

Message: You cannot select a table space of type LONG for this field.

Explanation: The table space cannot be of type LONG.

EWS2425

Message: You cannot alter the primary table space < *objectName* >.

Explanation: You cannot alter the primary table space. The database might be out of synch.

EWS2427

Message: You cannot search for this object until a system has been specified. Please go to the General page and make sure the system field is not blank.

Explanation: The object that you wish to search for must be in the same system as the object that you are creating. Therefore, a system must be chosen before you can perform the search. The system is determined by selecting a Relational Database Qualifier object on the General page.

EWS2428

Message: The qualifier you have selected cannot be used because it has a different system.

Explanation: Related objects must refer to the same system. There is at least one object that is related to the object you are creating. In order to change the system of the object you are creating, you will first have to detach all the related objects.

EWS2429

Message: Logged attribute of the LOB column changed in table < *objectName* >.

Explanation: Logged attribute of the LOB column changed. The database might be out of synch.

EWS2430

Message: Compact attribute of the LOB column changed in table < *objectName* >.

Explanation: Compact attribute of the LOB column changed. The database might be out of synch.

EWS2431

Message: The query *name* which is in file *.QRY file name* could not be added to the SQL Query Folder.

Explanation: While importing the definition for an SQL query, DataAtlas found the that name is already being used for the query in file *.QRY file name* .

To correct the problem, either delete the duplicate icon from the SQL Query Folder or rename the conflicting SQL Query. If you delete the icon, you can create a new SQL Query, and then use the Search button on the General Page to find the file *.QRY file name* .

EWS2432

Message: Errors were detected while trying to import some sample queries into the DataAtlas SQL Query Folder. The following errors were found:

error list

EWS2433

Message: You cannot add a volume until a system has been specified. Please go to the General page and make sure the system field is not blank.

Explanation: The volume must be in the same system as the storage group you are creating. Therefore, a system must be chosen before you can add any volumes. The system is selected on the General page.

EWS2434

Message: The volume you have selected cannot be used because it is in a different system.

Explanation: Related objects must refer to the same system. You cannot choose this volume because it is not in the same system as this storage group.

EWS2435

Message: One or more errors were detected during reconcile mapping table validation. Refer to the populate report or runtime log for details about the entry that needs to be corrected.

EWS2436

Message: Entry *entry* contains fewer than the required six fields. Entries must contain the required six fields (words). Fields may not be defaulted.

EWS2437

Message: The sixth word of entry *entry* must be a valid DataAtlas name of the form |<prefix>name<variation:revision>| where the prefix, variation, and revision are optional. Only target names of this form are supported.

EWS2438

Message: The fifth word of entry *entry* must be DSDataElement or DSDBD. Only one of these classes may be specified as the target class.

EWS2439

Message: The fourth word of entry *entry* must be the length of the source data item or column definition specified as exactly five numeric characters (with leading zeroes if necessary). Data item or column definition lengths must consist of five numeric characters.

EWS2440

Message: To determine data type value for a particular data item (DI) or column definition (CD), create a mapping table containing the DI or CD name of interest and populate the corresponding structure or table using the TRIAL RUN and RECONCILE options. The resulting report will show the data type (and length) of the DI or CD in DataAtlas terms. These values should be used as the third and fourth words, respectively, for the DI or CD.

EWS2441

Message: The third word of entry *entry* must be the data type of the source data item or column definition specified as exactly two numeric characters (with a leading leading zero if necessary). Only DataAtlas data types that can be specified as two numeric characters are supported.

EWS2442

Message: The second word of entry *entry* must contain 1-30 characters. You may either modify the second word so that the maximum name length is not exceeded or delete this entry.

EWS2443

Message: The first word of entry *entry* must be DSDatatItem or DSRColumnDefinition or DSDBD. You may either modify the first word to contain a supported source class or delete this entry.

EWS2444

Message: The reconcile mapping table does not contain any entries. Reconcile cannot function with an empty mapping table. You need to add appropriate mapping table entries to the table, or specify a different Reconcile mapping table.

EWS2449

Message: The *name1* , *name2* already exists in the list.

EWS2461

Message: No unique key columns were selected.

Explanation: You need to select one or more unique key columns in order to create an unique key.

EWS2462

Message: An unique key with the same name already exists. Please change the name and try again.

Explanation: Pick a name that is not in the Unique Keys List.

EWS2463

Message: The combination of columns in the unique key has already been defined either by the primary key or another unique key in the list.

Explanation: Add or remove one or more columns from the unique key.

EWS2464

Message: The composite key has reached the limit of 16 columns.

Explanation: The composite key cannot have more than 16 columns.

EWS2465

Message: The key cannot be found in the data structure.

Explanation: Please ask your local administrator.

EWS2466

Message: The pointer to the index cannot be found. The index cannot be opened.

Explanation: Please try again later or ask your local administrator.

EWS2467

Message: The primary key is a duplicate of one of the unique keys.

Explanation: You need to either modify the current primary key or modify the unique key that duplicates the primary key.

EWS2468

Message: An error occurs while attempt to save the table space information.

Explanation: Please ask your local administrator.

EWS2469

Message: For unique key columns, you cannot use *type* data type.

Explanation: Either LONG or LONG RAW data types are not allowed.

EWS2470

Message: The platform of the system selected does not match the platform of the object.

Explanation: Please ask your local administrator.

EWS2471

Message: The operating system access method is being changed from ISAM to VSAM. As a result, Primary and Overflow Size values on the DATASET page may no longer be valid. Each Size value must be a multiple of 512 for values less than 8192, and a multiple of 2048 for values up to 30720 in DBDs that use VSAM.

Explanation: Please consult IMS references.

EWS2472

Message: A TeamConnection error occurred while trying to refresh the object(s) after an unsuccessful delete attempt.

Explanation: After an unsuccessful delete attempt, DataAtlas tries to refresh the object(s) by retrieving new copies from TeamConnection. This resulted in a TeamConnection error < *tcerror* >. To ensure that the object(s) are refreshed accurately, you should shut DataAtlas down and bring it back up before attempting any further actions on these object(s).

EWS2474

Message: *name* is an invalid constraint name.

Explanation: Please consult reference books about the object name restrictions.

EWS2475

Message: The file < *fileName* > already exists. Do you want to overwrite it with this query?

Explanation: The file which you've specified in which to store the query already exists. If you choose to overwrite the file, the contents of that file will be replaced by this query.

EWS2476

Message: The file < *fileName* > already exists. Do you want to overwrite it with this query report?

Explanation: The file which you've specified in which to store the query report already exists. If you choose to overwrite the file, the contents of that file will be replaced by this query report.

EWS2479

Message: The specified system is for a later release than the catalog from which tables are being populated.

Explanation: Please specify a system that matches the release of the catalog.

EWS2481

Message: One or more DLLs required to execute a DB2 operation could not be dynamically loaded. If you have not installed IBM Database 2 Version 2.1 or higher, please do so prior to executing this DB2 operation. If you have installed this product, ensure that the DLLs can be found in your environment.

Explanation: An attempt was made to execute a DB2 operation that requires IBM Database 2 Version 2.1 or higher,   but either the DLL(s) could not be found or cannot be loaded from your environment. If you would like to proceed with this DB2 operation, ensure that the IBM Database 2 product is installed and the DLLs can be found in your environment and then try the operation again.

EWS2483

Message: During a high level language Populate, no Symbol records were found in the SYSADATA file produced by the compiler. This may mean the compiler you are using is not supported by DataAtlas or the compiler is not at the required CSD level. Make certain the compiler you are using is listed in the Supported Products list in the DataAtlas Help.

Explanation: If you need further assistance, please consult your local administrator.

EWS2484

Message: The length exceeds the maximum number of bytes allowed for LOB data types.

Explanation: If only the length is specified, the maximum value is 2,147,483,647. If K is specified, the maximum value for length is 2,097,152. If M is specified, the maximum value for length is 2,048. If G is specified, the maximum value for length is 2.

EWS2485

Message: Error in the definition of the new foreign key.

Explanation: The new foreign key could be invalid for one or more reasons. 1) You have not chosen a matching column for each of the columns in the primary key of the referenced table. If the current drop-down box doesn't contain a match, you need to create a matching column from the Definition page of this table. 2) This is a duplicate foreign key. The definition of the key is already defined by another foreign key. 3) Your foreign key has duplicate columns. You cannot define a column in a foreign key more than once.

EWS2486

Message: You need to enter a name for the constraint.

Explanation: Type the constraint name in the Name field and try again.

EWS2487

Message: An attempt to store the object failed because the name of a related object is already used in this release. Possible actions to take are:

1) Integrate other workareas

2) Refresh the current workarea

3) Link parts in the other workarea to this workarea

4) Change release to concurrent mode (after integrating workareas)

Consult TeamConnection documentation for details.

Explanation: *tcError*

EWS2488

Message: The file *fileName* already exists. Do you want to overwrite this file?

Explanation: You've chosen to store your data to a file that already exists. If you continue with this action, the contents of the file will be overwritten with your new data.

EWS2489

Message: The file *fileName* does not exist.

EWS2490

Message: You must specify values for all the fields in this window in order to start DataAtlas. If you cancel this window, DataAtlas will not be started.

Do you want to specify the necessary values so that DataAtlas can be started?

Explanation: Before the DataAtlas Main Window can be shown, you need to enter some initial values for the TeamConnection family, release, work area, and component you would like to use for default values. After you enter any nonblank value, you can proceed.

EWS2491

Message: Table definition *tdefName* already belongs to a relational design. DataAtlas will not allow this table definition to be added to the current relational design.

Explanation: If you want to add this table definition to the current relational design, you have to remove the table definition from the relational design it belongs to now.

EWS2497

Message: You should not store the object using the default object name.

Explanation: Enter a name other than the default name.

EWS2510

Message: The *lang* name specified in the name field cannot be a duplicate of another item under the same parent.

Explanation: Rename the item using a nonduplicate name so that the window can be accepted.

EWS2512

Message: The workfolder *name* could not be stored.

Explanation: An attempt was made to store changes to workfolder *name* but the store attempt failed. When you close this workfolder, you can attempt to store these changes again or close the folder without saving.

EWS2513

Message: No databases were found using the search pattern *pattern* .

Explanation: No databases were found that match the search criteria.

EWS2517

Message: The following objects could not be opened because they do not exist in the TeamConnection database:

object type, object name (version)

EWS2518

Message: The *name* Output file field is incorrectly specified.

Explanation: The field may include a directory that does not exist. Create the directory and click OK, or click Cancel to cancel the generate operation.

EWS2519

Message: The *name* Output directory does not exist.

Explanation: Create the directory and click OK, or click Cancel to cancel the generate operation.

EWS2520

Message: *msg*

Explanation: The field may include a directory that does not exist. Create the directory and click OK, or click Cancel to cancel the generate operation.

EWS2526

Message: The name you have entered for this object is not a valid name.

Explanation: The name may be invalid for one or more reasons: 1) it's a reserved word, 2) it includes invalid characters, 3) it's length exceeds the maximum length allowed, 4) it's an empty string.

EWS2527

Message: While processing a return code= *returncode* from the *compilertype* compiler, a problem was encountered with the listing file, *filename* . The listing file could not be opened. This situation may be the result of executing an empty command file.

Explanation: If you compiled using a command file, make sure it is not empty. Also, make sure the listing file is not in use. If the problem persists, call your IBM service representative for assistance.

EWS2529

Message: The table contains a column with an invalid data type. Open the column's notebook, change the its data type, and try again.

Explanation: The column may be invalid for one or more reasons: 1) a shareable data element that it was depending on was deleted, 2) it was populated incorrectly, 3) it had an internal error while being stored.

EWS2530

Message: An existing file has the same name as an output file on the following pages:

Explanation: NIL

EWS2531

Message: *msg*

Explanation: An existing file has the same name as a generate output file. Click OK to let generate replace the existing file, or rename the existing file and click OK. Click Cancel to end the generate operation.