

AppBuilder
By Magic Software Enterprises

Magic Software AppBuilder

Version 3.2

Information Model Reference Guide

Corporate Headquarters:

Magic Software Enterprises
5 Haplada Street,
Or Yehuda 60218, Israel
Tel +972 3 5389213
Fax +972 3 5389333

© 1992-2013 AppBuilder Solutions

All rights reserved.

Printed in the United States of America.

AppBuilder is a trademark of AppBuilder Solutions. All other product and company names mentioned herein are for identification purposes only and are the property of, and may be trademarks of, their respective owners.

Portions of this product may be covered by U.S. Patent Numbers 5,295,222 and 5,495,610 and various other non-U.S. patents.

The software supplied with this document is the property of AppBuilder Solutions and is furnished under a license agreement. Neither the software nor this document may be copied or transferred by any means, electronic or mechanical, except as provided in the licensing agreement.

AppBuilder Solutions has made every effort to ensure that the information contained in this document is accurate; however, there are no representations or warranties regarding this information, including warranties of merchantability or fitness for a particular purpose. AppBuilder Solutions assumes no responsibility for errors or omissions that may occur in this document. The information in this document is subject to change without prior notice and does not represent a commitment by AppBuilder Solutions or its representatives.

1. Information Model Reference Guide	2
1.1 Introduction to the Information Model	2
1.2 Entity Types	7
1.2.1 Common Properties of Entity Types	8
1.2.2 Entity Type Description	9
1.2.2.1 Application Configuration	10
1.2.2.2 Attribute	10
1.2.2.3 Bitmap	11
1.2.2.4 Bitmap Implementation	12
1.2.2.5 Business Object	13
1.2.2.6 Collection	14
1.2.2.7 Column	15
1.2.2.8 Component	17
1.2.2.9 Component Folder	18
1.2.2.10 Database	19
1.2.2.11 Data type	20
1.2.2.12 Drawing	22
1.2.2.13 Entity	23
1.2.2.14 Event	25
1.2.2.15 Field	26
1.2.2.16 File	29
1.2.2.17 Function	30
1.2.2.18 Identifier	31
1.2.2.19 Key	32
1.2.2.20 Logical Process	33
1.2.2.21 Machine	35
1.2.2.22 Migration	36
1.2.2.23 Partition	37
1.2.2.24 Physical Event	39
1.2.2.25 Process	40
1.2.2.26 Relationships	42
1.2.2.27 Report	43
1.2.2.28 Rule	45
1.2.2.29 Section	47
1.2.2.30 Server	48
1.2.2.31 Set	48
1.2.2.32 State	50
1.2.2.33 Symbol	51
1.2.2.34 Table	52
1.2.2.35 Transition	54
1.2.2.36 Value	55
1.2.2.37 View	55
1.2.2.38 Window	56
1.3 Relationship Types	57
1.3.1 Common Properties of Relationship Types	57
1.3.2 Reference of Relationship Types	59
1.4 Text Property Conventions	91
1.5 Enterprise Administration Information Model	91
1.6 Enterprise Repository Objects	99
1.7 Backwards Compatibility	139

Information Model Reference Guide

The AppBuilder Information Model is the collection of object types available to business planners, analysts, system designers, and developers who work in the AppBuilder environment. This guide provides information about the entity types and relationship types within the Information Model. These entity types and relationship types are listed alphabetically, and each listing includes a description about the object and its properties.

This guide includes the following sections and topics:

- [Introduction to the Information Model](#)
- [Entity Types](#)
- [Relationship Types](#)
- [Text Property Conventions](#)
- [Enterprise Administration Information Model](#)
- [Enterprise Repository Objects](#)
- [Backwards Compatibility](#)

Introduction to the Information Model

Introduction to the Information Model

The AppBuilder Information Model is the collection of object types available to business planners, analysts, system designers, and developers who work in the AppBuilder environment. This guide provides information about the entity types and relationship types within the Information Model. These entity types and relationship types are listed alphabetically, and each listing includes a description about the object and its properties.

Understanding the Information Model

The Information Model is a collection of entity types and relationship types, collectively called object types. This discussion includes the following:

- [Information Model Elements](#)
- [Accessing Properties](#)
- [Comparative Terminology](#)
- [Common Properties](#)
- [Diagrammatic Summary](#)

When you build an application, you tailor object types. When you assign a unique name to an object type, you create an object, which is an *instance* of that object type. For example, a File entity type designed to hold business addresses might be called CUSTOMER_ADDRESSES. You build models of your business, and you design and develop applications by defining particular instances of these object types, making them unique by defining their properties. Thus, you use object types to create objects.

Use nouns to represent the names of instances of entity types. For example, an instance of the Rule entity type that gets customer information from a file might be called CUSTOMER_FETCHER. Use a noun-verb-noun combination to represent instances of relationship types. For example, the relationship instance between the rule and the file just mentioned is CUSTOMER_FETCHER accesses CUSTOMER_ADDRESSES.

Information Model Elements

The elements of the Information Model include the following:

- [Entity Types](#)
- [Relationship Types](#)
- [Properties](#)
- [Repositories](#)
- [Submodels](#)
- [Internal Object Types](#)

Entity Types

An entity type is anything about which you can store data (for example, a table, a person, or a corporation). In Information Model diagrams, entity types appear as boxes, and their names appear as nouns.

Relationship Types

A relationship type indicates how an entity type interacts with another entity type. In Information Model diagrams, relationship types appear as

lines, and their names appear as verbs. Because a relationship type always connects two entity types, you can view the relationship from the perspective of either entity. For example, a Process is-defined-by a Rule or a Rule defines a Process are two different ways of looking at the same relationship.

Properties

Properties, or attributes, define both entity types and relationship types. In the Construction Workbench interface, Properties refer to properties of entity types, and Relationship Properties refer to properties of relationship types. There can be several specific properties of an object type. A property can be required, optional, or generated. If it is a required property, you must provide a value for it to save the instance successfully. If it is an optional property, you can save the instance without providing a value for it. If it is a generated property, the AppBuilder environment populates it, and you cannot alter it.

Repositories

The models and applications that you build are stored in repositories. There are three implementations of repositories in the AppBuilder Environment:

- Personal (local) repository
- Workgroup (server) repository
- Enterprise (mainframe) repository

The personal repository resides on a workstation with several development tools that you can use to build new or edit existing instances of repository object types. The personal repository stores the data that is accessed by all the tools.

The workgroup repository is a server-based or departmental repository that resides on a network server. Developers can simultaneously view and access any objects in this repository.

The enterprise repository resides on the mainframe (host). Developers can upload and download objects between this repository and a workstation Personal Repository. For more details about the enterprise repository, refer to the *Enterprise Administration Guide*.

Repositories store data about your business and the system you are building. All repositories use the same Information Model, though there are a few differences between the workstation and host objects (see [Backwards Compatibility](#)). For more details, refer to the *Repository Administration Guide for Workgroup and Personal Repositories*.

Submodels

The Information Model can be broken down into functional subsets called submodels. For example, the part of the Information Model that deals with security can be viewed as the Security submodel (see [Diagrammatic Summary](#) for more information).

All object types are part of the Information Model unless otherwise noted. Generally, this guide presents a property of an object type as it appears on the workstation, which might differ slightly from the way it appears on the host. For instance, properties that appear as check boxes on the workstation usually require you to choose between two items and perform a numeric selection on the mainframe.

Internal Object Types

There are a few entity and relationship types that the AppBuilder environment uses internally. These objects are generally invisible to you while you are working within the environment. However, you must separately upload and download these objects to and from the host; these objects are consequently visible within the repository maintenance tool and can be referred to during a Workgroup or Enterprise Repository migration. This manual lists these objects in the event that you need to identify them during host communications, but they are not otherwise available for your use.

Accessing Properties

To access the properties of entity type objects or relationship type objects, from the Construction Workbench, right-click objects in the Hierarchy window in any of the tabs: Project, Configuration, Repository, Inverted, and select Properties.

To access the properties from the Repository Administration, complete the following steps:

From the menu, select *Tools > Query Content*.

1. From the Query Repository Content window, select an Object type and click *Query*.
2. Select an Object instance and click *Insert*.
3. From the Browser window, right-click the object and choose *Properties*.

The Properties window displays with the following tabs:

- The *General* tab contains the editable properties of that entity.
- The *Audit* tab contains the common properties of that entity.
- The *Remote Audit* tab contains information about migration of the entity to and from the enterprise repository.
- The *Text* tab contains a space for you to enter descriptive text about that entity.
- The *Keywords* tab contains a space for you to enter a list of keywords to associate with that entity.

Comparative Terminology

The use of entity types in AppBuilder is significantly different from the use of their third-generation language (3GL) equivalents in a traditional development environment. [Information Model Terms Compared with 3GL Terms](#) compares some of the Information Model entity types to their 3GL equivalents. Use this table to bridge the gap between traditional terms used by programming languages, such as COBOL or PL/I, and equivalent terms used by AppBuilder Information Model entity types commonly used during system development, along with the closest corresponding term.

Information Model Terms Compared with 3GL Terms	
Traditional term	AppBuilder term
Menu item	Process
Paragraph or routine	Rule
Screen	Window
Subroutine	Component
Record	View
Field or variable	Field
File or table	File

Keep in mind that the correspondences are not exact equivalents. For example, the View entity type is not the same as a 3GL environment Record. In AppBuilder, a view constitutes a data definition, which you can attach to a file. In this case, a view would, in effect, define a record. You also can use the same view to define the output of a rule used to populate the file, or associate it with a window used to display the data.

Common Properties

All entity types have common properties. All common properties except System ID are the entity type Audit properties. The Audit properties, except for local date and local time, are not populated until the entity type has either been created on the host, either directly or by uploading, or committed to the workgroup repository.

In the enterprise repository on the host, *all* properties appear on the main screen for each object type and may appear in a different order.

In the personal and workgroup repositories on the workstation, Name and System ID, as well as all properties unique to each object type, appear on the General tab of the Properties dialog for that object type. Audit properties appear on the Audit tab of the entity type (see [Audit and Remote Audit Tabs](#)). Property names might be slightly different among the workstation and host repositories, and some properties might not appear in both (see [Backwards Compatibility](#)).

Audit and Remote Audit Tabs

Properties - [Rule] ✕

General | Audit | Remote Audit | Test | Keywords

Name: HELLO_HTML

Project: PFCE-SAMPLES

Owner: TEST

Last user: YSA:MI

Revision: 11

Version: 1.1.1.1.1.1

Date maintained: 2009-22

Time maintained: 13:54:3E

Locked by:

Properties - [Rule] ✕

General | Audit | Remote Audit | Test | Keywords

Name: HELLO_HTML

Owner: TEST

Project: PFCE-SAMPLES

Explicit lock:

QA status:

Change number: 11

CREATED: MAINTAINED

Remote date: 00/09/21 00/09/21

Remote time: 12:30 12:30

Remote user: TEST TEST

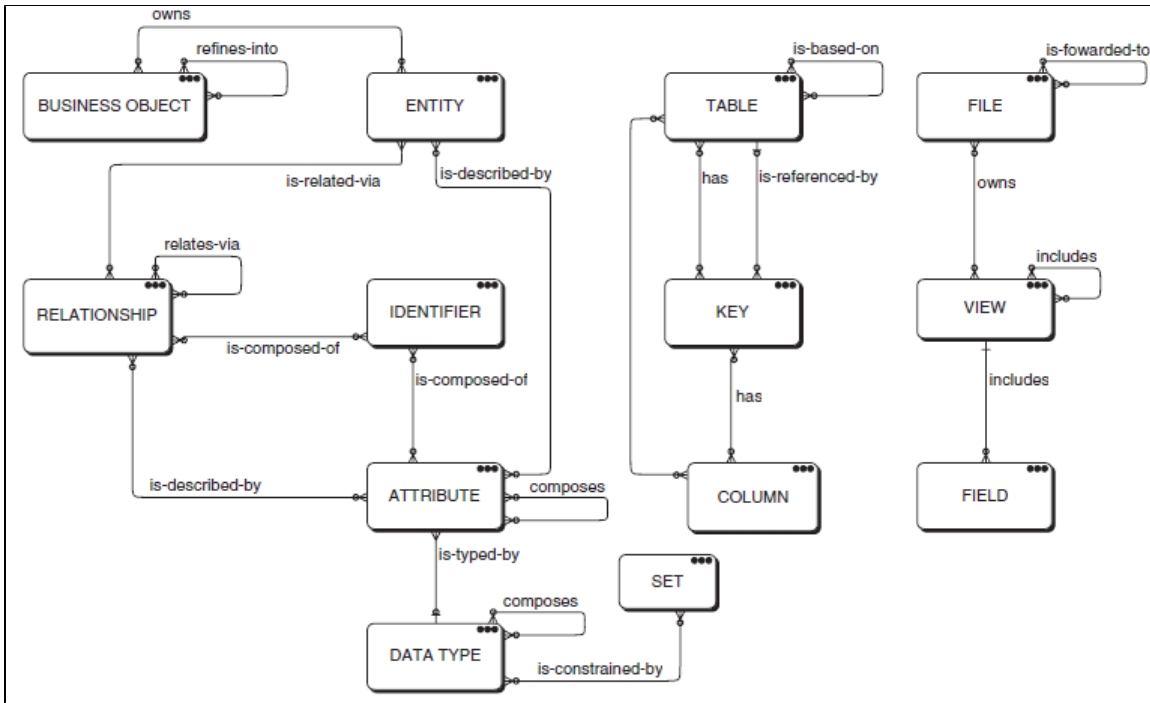
OK Cancel Apply Undo

Diagrammatic Summary

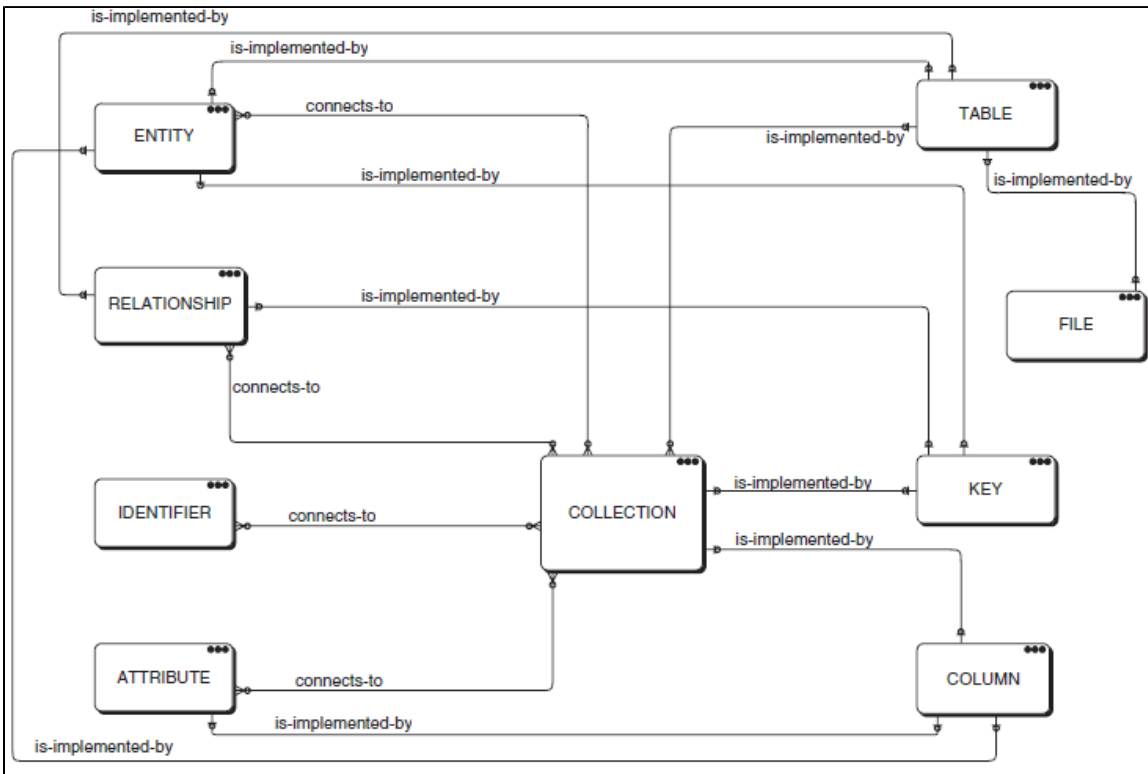
The following diagrams graphically illustrate many of the entity and relationship types described in this book. The Information Model is divided into the following views:

- [Design and Construction: Data View](#)
- [Design and Construction: Data Traceability View](#)
- [Design and Construction: Process View](#)
- [Security View](#)

Design and Construction: Data View



Design and Construction: Data Traceability View



Design and Construction: Process View

An object type is anything about which information can be stored in the AppBuilder repository. Object types have properties that describe them. There are two categories of object types in the repository: entity types and relationship types. This section lists properties common to all entity types and then provides a detailed description of each entity type. This chapter includes the following topics:

- [Common Properties of Entity Types](#)
- [Entity Type Descriptions](#)

Name and System ID properties are described in [Common Properties of Entity Types](#). Each section under [Entity Type Descriptions](#) includes a description of the entity type and how it is used, a screen shot of the Properties dialog, specifically the General tab where properties can be edited, a table listing its specific properties, and the relationships associated with the entity type.

Common Properties of Entity Types

Common Properties of Entity Types

The common properties of entity types in the AppBuilder Information Model include the following:

- [Name](#)
- [System ID](#)
- [Owner](#)
- [Project](#)
- [Explicit lock \(Personal or Workgroup\)](#)
- [Change number](#)
- [Created/Remote date \(Personal or Workgroup\)](#)
- [Date created \(Enterprise\)](#)
- [Created/Remote time \(Personal and Workgroup\)](#)
- [Time created \(Enterprise\)](#)
- [Created/Remote user \(Personal and Workgroup\)](#)
- [Created by \(Enterprise\)](#)
- [Maintained/Remote date \(Personal and Workgroup\)](#)
- [Date maintained \(Enterprise\)](#)
- [Maintained/Remote time \(Personal and Workgroup\)](#)
- [Time maintained \(Enterprise\)](#)
- [Maintained/Remote user \(Personal and Workgroup\)](#)
- [Maintained by \(Enterprise\)](#)
- [Maintained/Local date \(Personal and Workgroup\)](#)
- [Maintained/Local time \(Personal and Workgroup\)](#)
- [Version \(Enterprise only\)](#)

<i>Common properties of entity types</i>		
Property	Status	Description
Name	Required	<p>Unique name, up to thirty (30) characters in length, identifies each instance of an object type. Since names provide information about the function and purpose of an object instance, failure to adhere to a standard naming convention makes it hard to quickly locate and use an existing instance. See Text Property Conventions.</p> <p>Follow these general guidelines for creating names:</p> <ul style="list-style-type: none"> • Abbreviate. Use abbreviations where necessary. (See Abbreviation Guidelines.) • No Spaces. The name of an instance cannot contain spaces. To separate character strings in a name, use an underscore (_) between each word. • Object-Action. Names should describe the function or purpose of an instance by first describing what it is and then describing what it does. • Names are stored in uppercase. Take this into account when performing comparisons in the Rule source code.
System ID	Generated	<p>An alphanumeric string the AppBuilder environment uses internally to refer to the instance of the object type. Each instance has a unique System ID. During workstation installation, a workgroup repository System ID (a number between 400 and 1295) for your workstation was entered. The System ID of an instance is generated from this number. For the System IDs of an instance to be unique across an application, the Workgroup Repository System ID chosen at installation must be unique from all other developers of that application.</p>
Owner	Generated	<p>ID of the owner of the instance. Can be different from the ID generated for the created/remote user property.</p>

Project	Generated	Name of the project to which the instance belongs.
Explicit lock (Personal or Workgroup)	Generated	The field on the left side of the window indicates whether the instance is locked (Y) or not locked (N). If it is locked, the field on the right side of the window lists the ID of the user who locked it. If an instance is locked, you cannot change it in the enterprise repository either directly or by uploading.
Change number	Generated	System-generated number to protect instances of an object from simultaneous changes from two users. An instance can be uploaded only if its change number matches the change number in the enterprise repository.
Created/Remote date (Personal or Workgroup)	Generated	Date the instance was first created in the Personal or Workgroup Repository, either directly or by uploading.
Date created (Enterprise)	Generated	Format: YY/MM/DD Date the instance was first created in the enterprise repository, either directly or by uploading.
Created/Remote time (Personal and Workgroup)	Generated	Time the instance was first created in the Personal or Workgroup Repository, either directly or by uploading.
Time created (Enterprise)	Generated	Format: HH:MM Time the instance was first created in the enterprise repository, either directly or by uploading.
Created/Remote user (Personal and Workgroup)	Generated	ID of user who created the instance in the Personal or Workgroup Repository, either directly or by uploading.
Created by (Enterprise)	Generated	ID of user who created the instance in the enterprise repository, either directly or by uploading.
Maintained/Remote date (Personal and Workgroup)	Generated	Date the instance was last changed in the Personal or Workgroup Repository, either directly or by uploading.
Date maintained (Enterprise)	Generated	Format: YY/MM/DD Date the instance was last changed in the enterprise repository, either directly or by uploading.
Maintained/Remote time (Personal and Workgroup)	Generated	Time the instance was last changed in the Personal or Workgroup Repository, either directly or by uploading.
Time maintained (Enterprise)	Generated	Format: HH:MM Time the instance was last changed in the enterprise repository, either directly or by uploading.
Maintained/Remote user (Personal and Workgroup)	Generated	ID of the last user who changed the instance in the Personal or Workgroup Repository, either directly or by uploading.
Maintained by (Enterprise)	Generated	ID of last user who changed the instance in the enterprise repository, either directly or by uploading.
Maintained/Local date (Personal and Workgroup)	Generated	Format: YY/MM/DD Date the instance was last changed on the workstation.
Maintained/Local time (Personal and Workgroup)	Generated	Format: HH:MM Time the instance was last changed on the workstation.
Version (Enterprise only)	Generated	Identifies the logical repository in which this instance resides.

Entity Type Description

Entity Type Descriptions

The entity types in the AppBuilder Information Model (in alphabetical order) are shown in [Information Model Entity Types](#).

Information Model Entity Types		
---------------------------------------	--	--

Application Configuration	Event	Report
Attribute	Field	Rule
Bitmap	File	Section
Bitmap Implementation	Function	Server
Business Object	Identifier	Set
Collection	Key	State
Column	Logical Process	Symbol
Component	Machine	Table
Component Folder	Migration	Transition
Database	Partition	Value
Data Type	Physical Event	View
Drawing	Process	Window
Entity	Relationships	

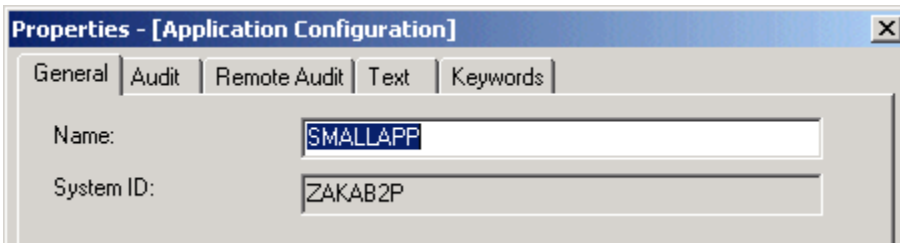
Refer to [Common Properties of Entity Types](#) for properties common to all entity types. Properties specific to each of the listed entity types are shown in the Properties Table under each entity heading. If there is no Properties Table, that indicates there are no properties specific to that entity type.

Application Configuration

Application Configuration

The Application Configuration entity type ([Application Configuration Properties Dialog](#)) encapsulates the information needed to prepare a client-server application, to migrate it to a production environment, and to administer it at runtime.

Application Configuration Properties Dialog



Child Relationships

Application Configuration has Partition

Current MRE AppBuilder model allows only one process per partition. If a Batch partition is to be uploaded and synchronized with the MRE, the application configuration and its partitions have to conform to the MRE model. Therefore, if there is more than one Batch root rule, multiple processes have to be created, one per root rule. And all of these processes will be attached to a partition object.

Parent Relationships

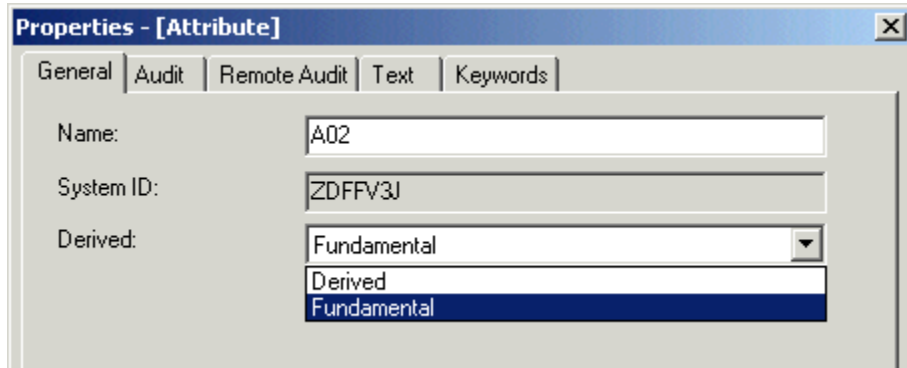
- Application Folder contains an Application Configuration
- Application Configuration partitions a Function. This is the inverse relationship added for a Function; the relationship already exists in the relationships list.

Attribute

Attribute

The Attribute ([Attribute Properties Dialog](#)) describes the characteristics or properties of an entity or a relationship. Typically, you define instances of Attributes during business object analysis to provide details for your data model. For example, if you have an object called Customer, you might create attributes called First_Name, Last_Name, Street_Address, and so on.

Attribute Properties Dialog



Attribute properties		
Field in Dialog	Property	Description
Derived	Attr_Derived	Status: Required Default is Fundamental mDomAttributeType The following choices are available: <ul style="list-style-type: none"> • Fundamental • Derived
N/A	DerivationFormula	Status: Optional Can contain a maximum of 240 characters

Child Relationships

- Attribute is customized to Attribute
- Attribute is composed of Attribute
- Attribute is composed of Relationship
- Attribute is typed by Data Type
- Attribute is dominated by Business Data Domain

Parent Relationships

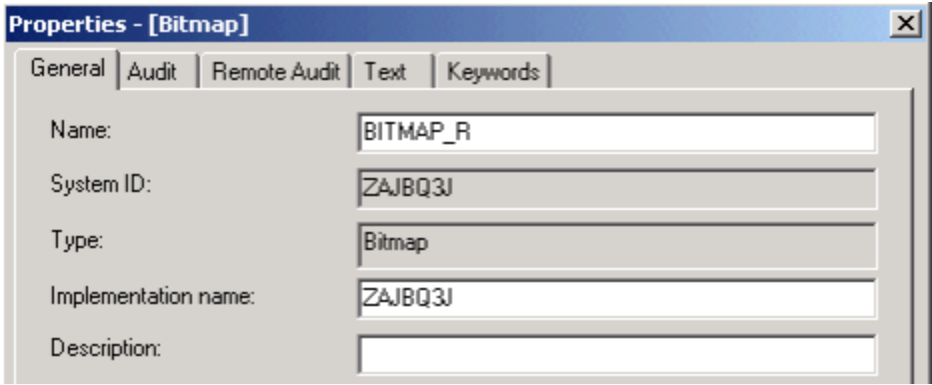
- Application Folder contains Attribute
- ER View involves Attribute
- Comm Flow transports Attribute
- Scheme maps to Attribute
- Value maps to Attribute
- Business Concept maps to Attribute
- Business Object is customized to Attribute
- Event is customized to Attribute
- Business Data Domain is customized to Attribute
- Relationship is customized to Attribute
- Entity is customized to Attribute
- Entity is described by Attribute
- Relationship is described by Attribute
- Attribute is composed of Attribute
- Identifier describes Attribute

Bitmap

Bitmap

The Bitmap entity type ([Bitmap Properties Dialog](#)) represents a bitmap. A bitmap object is used to add graphic interest to windows and to camouflage hot spot objects.

Bitmap Properties Dialog



Bitmap Properties		
Field in Dialog	Property	Description
Type	Bitmap_Type	Status: Optional Default is Bitmap mDomBmpType The following choices are available: <ul style="list-style-type: none"> • Bitmap • Icon • Pointer • JPEG
Implementation Name	Bitmap_Imp_Name	Status: Generated Can contain a maximum of 8 characters
Description	Description	Status: Optional Up to 30 characters of additional information

Child Relationships

Bitmap has Bitmap Implementation

Parent Relationships

- Application Folder contains Bitmap
- Function has Bitmap
- Process has Bitmap
- Report has Bitmap
- Rule has Bitmap
- Window has Bitmap

Bitmap Implementation

Bitmap Implementation

The Bitmap Implementation is the entity type that manages the physical files that can comprise a bitmap. It exists to facilitate uploading objects to the host.

Bitmap Implementation Properties	
Property	Description
CountryLanguage	Status: Optional Default is English (US) mDomCountryLanguage The following choices are available: <ul style="list-style-type: none"> • English (US) • English (UK)

X_Res	Status: Optional Can contain a maximum of 5 characters
Y_Res	Status: Optional Can contain a maximum of 5 characters
FormatType	Status: Optional Default is BMP mDomBMPFormatType The following choices are available: <ul style="list-style-type: none"> • BMP • ICR • PTR • JPEG • GIF • WAV • AVI • MOV • MPEG • QTW
Description	Status: Optional Up to 30 characters of additional information that describes the bitmap

Parent Relationships

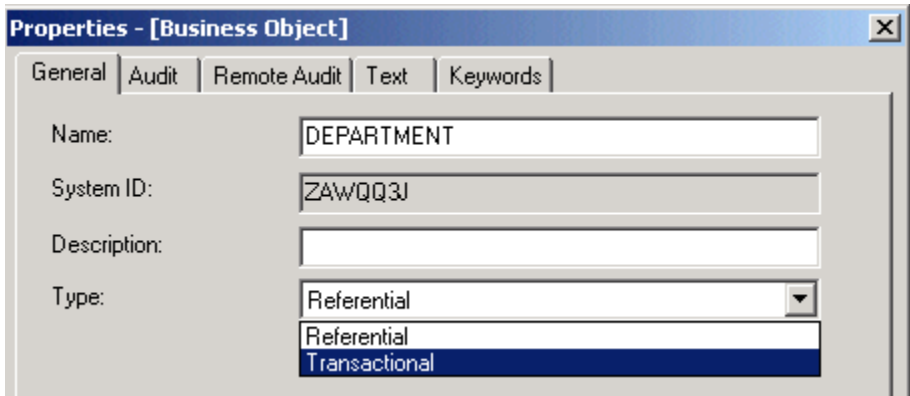
Bitmap is-had-by Bitmap Implementation

Business Object

Business Object

The Business Object ([Business Object Properties Dialog](#)) groups together a set of entities.

Business Object Properties Dialog



Business Object Properties		
Field in Dialog	Property	Description
Description	Description	Status: Optional Up to 30 characters of additional information that describes the Business Object
Type	BusObj_Type	Status: Required Default is Referential mDomBusinessObjectType Designates the type of Business Object. The following choices are available: <ul style="list-style-type: none"> • Referential • Transactional

Child Relationships

- Business Object data-content-defined-by ER View
- Business Object has State
- Business Object has-STD Drawing
- Business Object owns Entity
- Business Object owns Event
- Business Object owns Business Object
- Business Object owns Logical Process
- Business Object refines-into Business Object
- Business Object has member Entity
- Business Object has member Class Property
- Business Object has member Class Domain
- Business Object has member Class Object
- Business Object has member Class Relationship

Customized_To Relationships		
FROM		TO
Logical Business Function	Customized	Logical Process
Logical Workflow		Entity
Business Object		Attribute
Attribute		Critical Business Process
Logical Process		Logical Workflow
Business Data Domain		Logical Business Function
Event		Business Object
Relationship		Business Data Domain
Critical Business Process		Relationship
Entity		Event

Parent Relationships

- Event influences Business Object
- Application contains Business Object
- Scheme maps to Business Object
- Value maps to Business Object
- Business Concept maps to Business Object
- Logical Business Function maps to Business Object
- Model Scope includes Business Object
- Event influences Business Object

Collection

Collection

The Collection entity type ([Collection Properties Dialog](#)) stores internal traceability information. You can view this entity type, but do not modify it.

Collection Properties Dialog

The screenshot shows a dialog box titled "Properties - [Cell]". It has several tabs: "General", "Audit", "Remote Audit", "Text", and "Keywords". The "General" tab is active. Inside the dialog, there are three input fields: "Name" with the value "ACELL", "System ID" with the value "ZDFTF3J", and "Protocol" with a dropdown menu showing "LU62".

Collection Properties

Property	Description
Type	Status: Optional The type of the Collection. Maximum of 3 characters.
Nature	Status: Required The nature of the Collection. The following choices are available: <ul style="list-style-type: none"> • Logical • Physical • Relational
Number of elements	Status: Optional The number of elements in the Collection. This can be a numeric Value from 0 to 99.
Number of groups	Status: Optional The number of groups in the Collection. This can be a numeric Value from 0 to 99.

Child Relationships

- Collection connects-to Collection
- Collection connects-to Column
- Collection connects-to Key
- Collection connects-to Table
- Collection is-implemented by Collection
- Collection is-implemented by Column
- Collection is-implemented by File
- Collection is-implemented by Key
- Collection is-implemented by Table
- Collection implements Collection

Parent Relationships

- Attribute connects-to Collection
- Attribute is-implemented-by Collection
- Collection is-connected-to Collection
- Collection is-implemented-by Collection
- Data type connects-to Collection
- Data type is-implemented-by Collection
- Entity connects-to Collection
- Entity is-implemented-by Collection
- Identifier connects-to Collection
- Identifier is-implemented-by Collection
- Relationship connects-to Collection
- Relationship is-implemented-by Collection
- Table is-implemented-by Collection

Column

Column

The Column entity type ([Column Properties Dialog](#)) represents a column in a database.

Column Properties Dialog

Properties - [Column] [X]

General | Audit | Remote Audit | Text | Keywords

Name: MDB_AMT

System ID: ZAP93NH

Column format: Decimal

Column length: 00015

Column fraction: 02

Average length: 00015

Implementation name: MDB_AMT

Column Properties		
Field in Dialog	Property	Description
Implementation name	Col_Imp_Name	Status: Optional Can contain a maximum of 30 characters Maximum of 30 alphanumeric characters (18 for DB2 and DBM Databases), the first of which must be alphabetic.
Column format	Col_Type	Status: Required Default is Character The following choices are available: <ul style="list-style-type: none"> • Boolean Object Reference (not implemented) • Character • Date • Decimal • Graphic Character (DBCS) • Image • Mixed Character • Picture • Small Integer or Integer • Text • Time • Timestamp • VarChar
Column length	Col-Length	Status: Generated if column format is Date, Time, or Timestamp. Required for any other column format. Can contain a maximum of 5 characters Default is Character How large the column is. The value range for this property depends on the column format, as shown in Minimum and Maximum Column Lengths (according to column format) .
Column fraction	Col_Scale	Status: Optional Can contain a maximum of 2 characters Default is 0 Number of decimal places. For decimal and picture Column Formats, the Column fraction can range from zero to the Column length. For other Column Formats, this property is irrelevant.
Average length	Col_Avg_Length	Status: Optional Can contain a maximum of 5 characters The average length of the Column. This can be a numeric Value from 0 to 99,999.

Minimum and Maximum Column Lengths (according to column format)		
Column Format	Minimum length	Maximum length

Character	1	4,000
Date	4	4
Decimal	1	31
Graphic Character (DBCS)	1	2,000
Mixed Character	1	4,000
Picture	1	30
Small Integer or Integer	Small: 15; Integer: 31	Small: 15; Integer: 31
Time	4	4
Timestamp	12	12
VarChar	1	4,000

Parent Relationships

- Attribute connects-to Column
- Attribute is-implemented-by Column
- Collection connects-to Column
- Collection is-implemented-by Column
- Data type connects-to Column
- Data type is-implemented-by Column
- Entity connects-to Column
- Entity is-implemented-by Column
- Identifier connects-to Column
- Identifier is-implemented-by Column
- Key has Column
- Relationship connects-to Column
- Relationship is-implemented-by Column
- Table has Column
- Table is-implemented-by Column

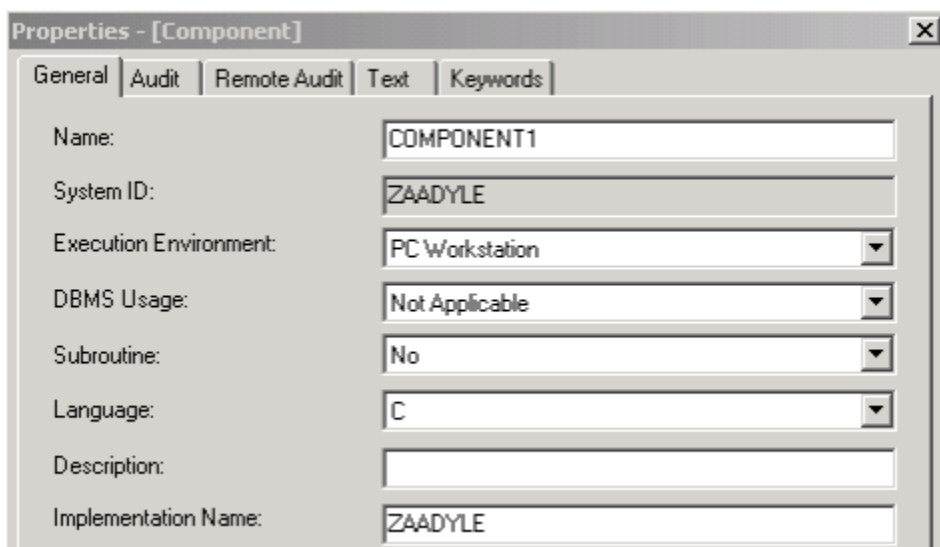
Component

Component

The Component entity type ([Component Properties Dialog](#)) contains code written in a third-generation computer language, such as C, COBOL, assembler, or PL/I, to do things that the Rules Language either cannot do or cannot do efficiently. This might be a complicated arithmetic algorithm (such as a square root), non-SQL data access logic (such as an IMS Database interface), or hardware-specific functions.

Components also support the concept of reengineering, where existing applications can be defined to the repository. Because components are written for a specific processing environment, they are not portable between environments.

Component Properties Dialog



Properties - [Component]

General | Audit | Remote Audit | Text | Keywords

Name: COMPONENT1

System ID: ZAADYLE

Execution Environment: PC Workstation

DBMS Usage: Not Applicable

Subroutine: No

Language: C

Description:

Implementation Name: ZAADYLE

Component Properties		
Field in Dialog	Property	Description
Description	Component	Status: Optional Up to 30 characters of additional information that describes the Component.
DBMS usage	DBMS_Usage	Status: Required Can contain a maximum of 4 characters Default is N/A mDomCompDBMSUsage Defines if the Component accesses a Database. The following choices are available: <ul style="list-style-type: none"> • DB2 • DL/I • N/A
Execution environment	Exec_Environ	Status: Required Can contain a maximum of 6 characters Default is PC (PC Workstation) mDomCompExecEnv The following choices are available: <ul style="list-style-type: none"> • IBM Mainframe Batch • PC User Component • PC Workstation • IBM Mainframe (CICS) • IBM Mainframe (IMS) • IBM Mainframe (CICS & Batch) • PC System Component
Implementation name	Comp_Imp_Name	Status: Optional Maximum of 8 alphanumeric characters, the first of which must be alphabetic.
Language	Language	Status: Required Can contain a maximum of 8 characters Default is C mDomLanguage The Language the Component is written in. The following choices are available: <ul style="list-style-type: none"> • Assembler , C, COBOL, PL/I, JAVA, PTAL
Subroutine	Exec_Mode	Status: Required Can contain a maximum of 4 characters Default is N mDomExecutionMode Defines if the Component is a subroutine. The following choices are available: <ul style="list-style-type: none"> • Yes • No
Sys_Source	Sys_Source	Status: Required Can contain a maximum of 3 characters Default is N/A

Child Relationships

- Component is-accessed-by File
- Component owns View
- Component refers-to Set
- Component uses Component

Parent Relationships

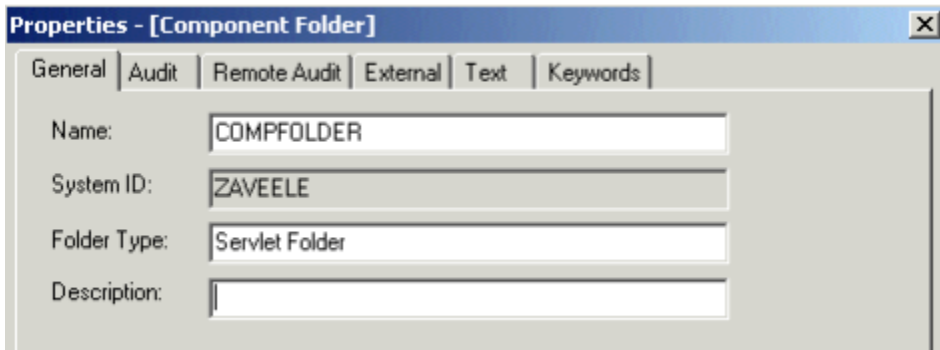
- Component is-used-by Component
- Partition encapsulates Component
- Rule uses Component

Component Folder

Component Folder

The Component Folder is a storage mechanism for objects (like files) that are not generated by AppBuilder. These are objects that do not fall into the usual object categories but the objects can be described (embodied) by file representations. Such objects are now represented within the repository using a Component Folder object. Each Component Folder represents one non-AppBuilder object.

Component Folder Properties Dialog



Component Folder Properties	
Property	Description
Folder Type	Status: Optional Can contain a maximum of 30 characters
Description	Status: Optional Can contain a maximum of 60 characters

Child Relationship

Component Folder content is Component Folder File

Parent Relationships

- Application Folder contains Component Folder
- Function has Component Folder
- Process has Component Folder
- Report has Component Folder
- Component has Component Folder
- Window has Component Folder
- Rule has Component Folder

Database

Database

The Database entity type ([Database Properties Dialog](#)) represents a database in a network configuration.

Database Properties Dialog

Properties - [Database]

General | Audit | Remote Audit | Text | Keywords

Name: ALLDB

System ID: ZAKUV2P

Machine name:

Type: DB2

Database directory path:

Implementation name:

Database Properties		
Field in Dialog	Property	Description
Machine name	Machine_Name	Status: Required The Machine associated with the Database in a Partition. Maximum of 30 characters.
Type	DBMS_Type	Status: Required Default is DB2 mDomDBMStype The Database type. The following choices are available: <ul style="list-style-type: none"> • DB2 • Oracle • MS-SQLServer • DB2/UDB
Database directory path	DB_Imp_Direct	Status: Required The runtime directory path. Maximum of 100 characters.
Implementation name	DB_Imp_Name	Status: Optional Can contain a maximum of 30 characters

Child Relationships

- Database contains Table
- Database is-accessed-by File
- Database is-related-to Database

Parent Relationships

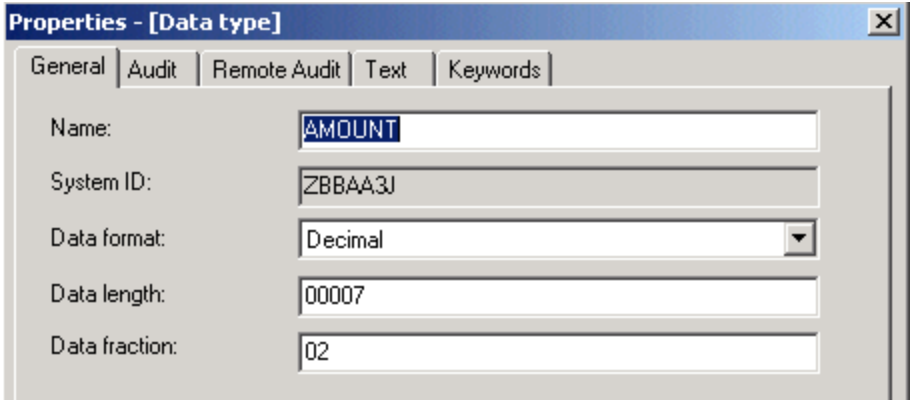
- Partition encapsulates Database
- Database relates-to Database

Data type

Data Type

The Data Type entity type ([Data Type Properties Dialog](#)) records a physical description of data. Typically, you define instances of Data Type entity types during business object analysis when you build your data model.

Data Type Properties Dialog



Data Type Properties	
Property	Description
Data Format	<p>Status: Required Default is Character The following choices are available:</p> <ul style="list-style-type: none"> • Boolean Object Reference (not implemented) • Character • Date • Decimal • Graphic Character (DBCS) • Image • Mixed Character • Picture • Small Integer or Integer • Text • Time • Timestamp • VarChar
Data Length	<p>Status: Generated if column format is Date, Time, or Timestamp. Required for any other column format. Can contain a maximum of 5 characters Default is Character How large the column is. The value range for this property depends on the column format, as shown in Minimum and Maximum Field Lengths (according to data format).</p>
Data Fraction	<p>Status: Optional Can contain a maximum of 2 characters Default is 0 Number of decimal places. For decimal and picture Column Formats, the Column fraction can range from zero to the Column length. For other Column Formats, this property is irrelevant.</p>

Minimum and Maximum Field Lengths (according to data format)		
Data Format	Minimum length	Maximum length
Character	1	4,000
Date	4	4
Decimal	1	31
Graphic Character (DBCS)	1	2,000
Mixed Character	1	4,000
Picture	1	30
Small Integer or Integer	Small: 15; Integer: 31	Small: 15; Integer: 31
Time	4	4

Timestamp	12	12
VarChar	1	4,000

Child Relationships

- Data Type composes Data Type
- Data Type connects-to Collection
- Data Type connects-to Column
- Data Type connects-to Key
- Data Type connects-to Table
- Data Type is-constrained-by Set
- Data Type is-implemented-by Collection
- Data Type is-implemented-by Column
- Data Type is-implemented-by File
- Data Type is-implemented-by Key
- Data Type is-implemented-by Table

Parent Relationships

- Attribute is-typed-by Data Type
- Data Type is-composed-of Data Type

Drawing

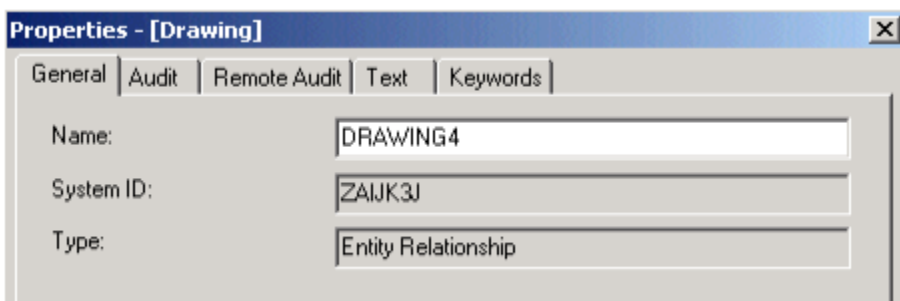
Drawing

The Drawing entity type ([Drawing Properties Dialog](#)) is a graphic representation of a user-defined model built of AppBuilder entity and relationship types. Drawing entity types are created on the workstation with the following tools:

- Entity-Relationship Diagrammer
- Process Dependency Diagrammer
- State Transition Diagrammer
- Matrix Builder
- Window Flow Diagrammer
- Database Diagrammer

Each Drawing entity has a drawing file with references to the object types in the drawing, but not to their definition. Drawing entity type relationships may not be viewed outside of these environments.

Drawing Properties Dialog



Drawing Properties		
Field in Dialog	Property	Description
Type	Drawing_Type	<p>Status: Generated</p> <p>Can contain a maximum of three characters</p> <p>Default is State Transition</p> <p>mDomDrawing</p> <p>Drawings are diagrams that can be reused.</p> <p>They are created with Construction Workbench tools.</p> <p>The drawing type indicates the tool used to make the drawing.</p> <p>For example, the Entity Relationship Diagram tool creates entity-relationship diagrams.</p> <p>The values for the types of drawings are summarized in Drawing description and values.</p>

Drawing entity type	
Property	Description
Name	Status: Required Common Property
System ID	Status: Generated Common Property
ShortName	Status: Optional Can contain a maximum of eight characters
Drawing_Type	Status: Generated Can contain a maximum of three characters Default is State Transition mDomDrawing Drawings are diagrams that can be reused. They are created with Construction Workbench tools. The drawing type indicates the tool used to make the drawing. For example, the Entity-Relationship Diagrammer creates entity-relationship diagrams. The following table shows the values for the types of drawings, as shown in Drawing description and values .

Drawing description and values	
Description	Values
State Transition	001
Entity Relationship	002
Process Dependency	003
Window Flow	010
Database Diagram	011
Physical Network	012
Data Store vs Entity	013
Function vs Entity	014
Function vs Location	015
Entity vs Location 1	016
Entity vs Location 2	017
Organization vs Entity	018
Process vs Entity	019
Organization vs Process	020
Matrix	021
System vs Process	022
Class Interaction	023

Parent Relationships

- Business Object has-a-state-transition Drawing
- Entity has-a-entity-relationship Drawing
- Entity has-a-state-transition Drawing
- Event has-a-process-dependency Drawing
- Logical Process has-a-process-dependency Drawing
- State has-a-state-transition Drawing

Entity

Entity

Entity

The Entity entity type ([Entity Properties Dialog](#)) describes the data an enterprise or organization uses. Typically, you define instances of entity object types during business object analysis when you build the data model. For example, an enterprise that rents automobiles might have a customer entity, a reservation entity, and a rental location entity.

Entity Properties Dialog

The following entity types are available:

- *Kernel* — A kernel entity is a basic entity that can exist independently from other entities in a logical model. This does not imply that it cannot have a relationship with other objects; rather, it implies the entity must be a unique object with its own Identifier. A kernel entity should not need relationships with other entities to be identified.
- *Associative* ---An associative entity associates two or more kernel or characteristic entities. Associative entities contain non-key properties and can be used to resolve many-to-many relationships.
- *Characteristic* ---A characteristic entity is a weak or dependent entity because it requires the existence of another entity in a logical model. Characteristic entities describe a kernel entity.
- *Intersection* ---An intersection entity associates two or more kernel entities and contains no properties. An intersection entity is the primary way to resolve many-to-many relationships between entities.

Entity Properties		
Field in Dialog	Property	Description
Type	Ent_Type	Status: Required Can contain a maximum of twenty characters Default is Kernel mDomEntityType The following choices are available:

- Kernel
- Associative
- Characteristic
- Intersection]

Expected minimum rows	Exp_Min_Rows	Status: Optional Can contain a maximum of five characters User-defined lower limit of the expected size of the instance (number of Fields). This can be a numeric Value from 0 to 99,999.
Expected maximum rows	Exp_Max_Rows	Status: Optional Can contain a maximum of ten characters User-defined upper limit of the expected size of the instance (number of Fields). This can be a numeric Value from 0 to 9,999,999,999.

Activity period	Exp_Duration	Status: Required Can contain a maximum of five characters Default is Day mDomDurarion Time period that describes the frequency of various operations performed with instances of this entity type. The choices are:
-----------------	--------------	---

- Day
- Week
- Month
- Year
- Quarter|

Avg. inserts for period	Ave_Wkly_Inserts	Status: Optional Can contain a maximum of four characters Estimated number of times during implementation that instances of this entity type will be created. This can be a numeric value from 0 to 9,999.
Avg. deletes for period	Ave_Wkly_Deletes	Status: Optional Can contain a maximum of four characters Estimated number of times during implementation that instances of this entity type will be deleted. This can be a numeric value from 0 to 9,999.
Avg. updates for period	Ave_Wkly_Updates	Status: Optional Can contain a maximum of four characters Estimated number of times during implementation that instances of this entity type will be updated. This can be a numeric value from 0 to 9,999.
Expected rows	Exp_Rows	Status: Optional Can contain a maximum of ten characters The number of instances of this entity type expected when you define the data model. This can be a numeric value from 0 to 9,999,999,999.
N/A	AveGrowthRate	Status: Optional Can contain a maximum of four characters

Child Relationships

- Entity connects-to Collection
- Entity connects-to Column
- Entity connects-to Key
- Entity connects-to Table
- Entity has-a-entity-relationship Drawing
- Entity has-a-state-transition Drawing
- Entity has State
- Entity has Identifier
- Entity is-accessed-by Project
- Entity is-described-by Attribute
- Entity is-implemented-by Collection
- Entity is-implemented-by Column
- Entity is-implemented-by Key
- Entity is-implemented-by Table
- Entity is-modified-by Development Project
- Entity is-modified-by Logical Process
- Entity is-related-via Relationship
- Entity owns View

Parent Relationships

- Assumption is-supported-by Entity
- Business Object owns Entity
- ER View involves Entity
- Function intersects-with Entity
- Goal is-supported-by Entity
- Information Need is-supported by Entity
- Logical Process affects Entity
- Problem affects Entity
- Problem is-caused by Entity
- Process impacts Entity
- Success Factor is-supported-by Entity

Event

Event

The Event entity type ([Event Properties Dialog](#)) represents an incident that acts as a stimulus to a business, system or object. It usually causes some activity or processing to be undertaken and may change the state of objects within the business or system. Events can be classified as external, internal or temporal. Examples are Customer_Calls, Accident_Reported, or Customer_Enters.

Event Properties Dialog

Event Properties		
Field in Dialog	Property	Description
Description	Event_Description	Status: Optional Up to thirty characters of additional information that describes the event.
Event type	Event_Type	Status: Optional Up to 10 characters of additional information that describes the type of event.
Event class	Event_Class	Status: Required Can contain a maximum of two characters Default is External mDomEventClass The choices are: External, Internal, Temporal

Child Relationships

- Event causes Transition
- Event composes Event
- Event is-composed-of Event
- Event has-a-process-dependency Drawing
- Event influences Business Object
- Event triggers Process
- Event triggers Logical Process

Parent Relationships

- Logical Process initiates Event
- Business Object owns Event

Field

Field

The Field entity type ([Field Properties Dialog](#)) records information about part of a file, such as a column in a DB2 Table, or part of the input or output definition of other entity types, such as the view owned by a window. The properties of a field describe the type and size of the data it contains.

The Field entity type is where you store the definition for each of the fields your application uses. Use the properties of the Field entity type to specify particular qualities for the data that can populate that field, such as integers, characters, and decimals. Also, see [Includes \(is-included-in\)](#), and [View](#).

Field Properties Dialog

Properties - [Field] [X]

General | Audit | Remote Audit | Text | Keywords

Name: INPUT_FIELD

System ID: ZAKAF2P

Field picture-storage:

Field picture-display:

Screen literal-long:

Field format: Character

Field length: 00015 Field fraction: 00

Range-minimum value:

Range-maximum value:

Reference table name:

Screen literal - short:

Implementation name: SMALL_INPUT_FIELD

Field Properties		
Field in Dialog	Property	Description
Field picture-storage	DB_Pic	Status: Required if Field format is Picture Status: Irrelevant for other Field formats Form in which the Field's Value should be stored, using the standard COBOL editing characters S (signed number), V (virtual decimal point), and 9 (numeric Value).
Field picture-display	Screen_Pic	Status: Optional if Field format is Picture Status: Irrelevant for other Field Formats Displays the Field Values as shown in COBOL Editing Characters and Field Display Forms , using the standard COBOL editing characters. <i>Note</i> : A picture Field can accept thirty display characters, but only eighteen of them can be numeric Value ("9") characters. This is a COBOL limitation.
Screen literal-long	Screen_Lit	Status: Optional Static text Window Painter and Report Painter use as a Field label when the Field is pasted onto a window. Maximum of twenty-four characters.
Field format	Field_Type	Status: Required Default is Character mDomFieldType The Field's data type. The following choices are available: <ul style="list-style-type: none"> • Boolean Object Reference • Character • Date • Decimal • Graphic Character (DBCS) • Image • Mixed Character • Picture • Small Integer or Integer • Text • Time • Timestamp • VarChar

Field length	Fld_Len	Status: Generated if Field Format is Date, Picture, Time, or Timestamp Status: Required for any other Field Format Can contain a maximum of five characters Default is fifteen The Value range for this Attribute depends on the Field Format, as shown in Minimum and Maximum Field Lengths (according to field format) .
Field fraction	Fld_Frac	Status: Generated if Field Format is Picture Status: Required if Field Format is Decimal Status: Irrelevant for any other Field Format Can contain a maximum of two characters Default is zero Number of decimal places in the Field. For Decimal Fields, the Field fraction must be greater than or equal to 0 and less than or equal to the Field length.
Range-minimum value	Field_Min	Status: Optional if Field Format is Decimal, Integer, Small Integer, or Picture Status: Irrelevant for other formats Can contain a maximum of 30 characters The minimum value that can be stored in this field.
Range-maximum value	Field_Max	Status: Optional if Field Format is Decimal, Integer, Small Integer, or Picture Status: Irrelevant for other formats Can contain a maximum of thirty characters The maximum Value that can be stored in this field.
Reference table name	Field_Ref	Status: Optional Can contain a maximum of eight characters System ID of the instance of the set entity type associated with the field and that lists acceptable inputs.
Screen literal-short	Screen_Lit_Short	Status: Optional Up to twelve characters of static text that report Painter uses as a field's short screen literal property.
Implementation name	Field_Imp_Name	Status: Optional Name that becomes the DB2 column name. It is limited to 18 characters and must conform to the naming standards for the application's database.

COBOL Editing Characters and Field Display Forms	
COBOL editing characters	Field display form
9	numeric value
*	check protection
,	comma
+	plus sign
-	minus sign
\$	floating currency symbol
.	decimal point
S	sign
Z	leading zero suppression
B	embedded blank
CR	credit
DB	debit

Minimum and Maximum Field Lengths (according to field format)		
Field format	Minimum length	Maximum length
Character	1	4,000
Date	4	4

Decimal	1	31
Graphic character (DBCS)	1	2,000
Mixed character	1	4,000
Picture	1	30
Small integer or integer	Small: 15; Integer: 31	Small: 15; Integer: 31
Time	4	4
Timestamp	12	12
VarChar	1	4,000

Child Relationships

- Field has Help Text
- Field refers-to Set
- Field uses Language

Parent Relationships

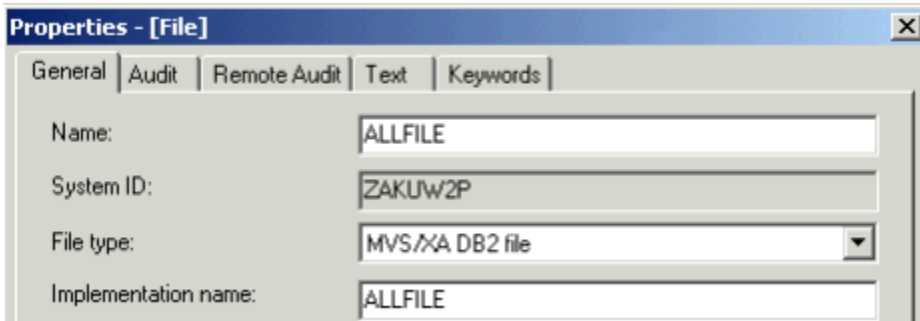
- File is-keyed-by Field
- View includes Field

File

File

The File entity type ([File Properties Dialog](#)) represents a physical data file on a disk. Rules and components can read from and write to disk files whose file entity types they are related to. Files can be automatically generated by AppBuilder tools.

File Properties Dialog



The screenshot shows a dialog box titled "Properties - [File]". It has five tabs: "General", "Audit", "Remote Audit", "Text", and "Keywords". The "General" tab is selected. Inside the dialog, there are four labeled fields: "Name:" with the value "ALLFILE", "System ID:" with the value "ZAKUW2P", "File type:" with a dropdown menu showing "MVS/XA DB2 file", and "Implementation name:" with the value "ALLFILE".

File Properties		
Field in Dialog	Property	Description

File type	File_Type	Status: Required Can contain a maximum of six characters Default is DB2 mDomFileType The following value is available in the list box: <ul style="list-style-type: none"> • XA DB2 File • DBM • Informix • Sybase • Oracle • DB2/2 • DB2/6000 • Tandem • Illustra • Ingres • DB2/400 • MS-SQLServer • DB2/UDB
Implementation name	File_Imp_Name	Status: Required Name that becomes the DB2 Table name. It is limited to thirty characters and must conform to the naming standards for the application database.

Child Relationships

- File is-forwarded-to File
- File is-keyed-by Field
- File owns View

Parent Relationships

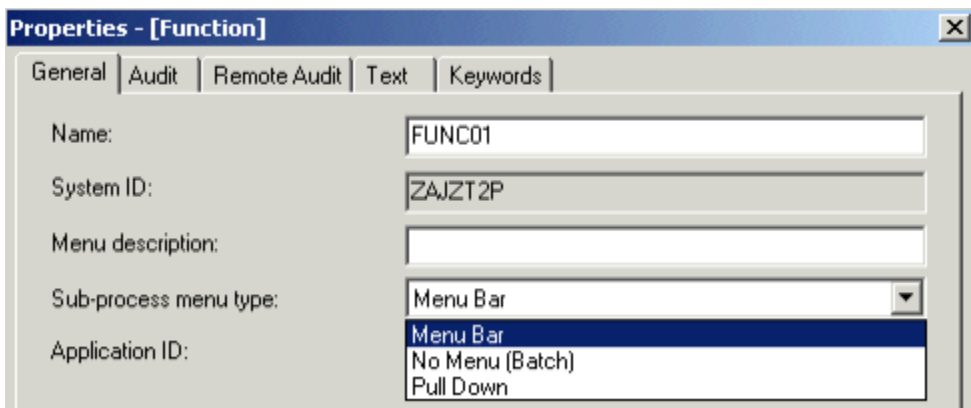
- Attribute is-implemented-by File
- Collection is-implemented-by File
- Component is-accessed-by File
- Partition encapsulates File
- Data type is-implemented-by File
- Database is-accessed-by File
- Entity is-implemented-by File
- Identifier is-implemented-by File
- Relationship is-implemented-by File
- Rule is-accessed-by File
- Table is-implemented-by File

Function

Function

The Function entity type ([Function Properties Dialog](#)) represents one of the major classes of activities for an enterprise, such as leasing, sales, and inventory control. Typically, you define instances of function entity types when you build the process model during analysis. In the execution environment, function names appear on the start-up menu bar. A function typically decomposes into one or more processes that can further decompose into additional processes.

Function Properties Dialog



Function Properties		
Field in Dialog	Property	Description
Menu description	Menu_Desc	Status: Required Up to thirty characters of text that appears as the first level of the menu bar.
Sub-process menu type	Child_Menu	Status: Required Default is Menu Bar mDomChildMenu Indicates how the processes under this function appear in the interface. The choices are: <ul style="list-style-type: none"> • Menu Bar • Pull-Down • No menu (Batch)
Application ID	Work_Station_Group	Status: Optional Up to eight characters that identify a group of applications grouped under this function.

Child Relationships

- Function has Bitmap
- Function intersects-with Entity
- Function refines-into Process
- Function partitioned by Application Configuration; the relationship already exists in the list of relationships

Parent Relationships

- Partition encapsulates Function

Identifier

Identifier

The Identifier entity type ([Identifier Properties Dialog](#)) is a logical key that becomes a physical key during database design. An identifier consists of one or more properties that uniquely identify an instance of a parent entity or cross-reference another entity.

Identifier Properties Dialog

Identifier Properties		
Field in Dialog	Property	Description
Type	Ident_Type	Status: Required Default is Primary mDomIdentifierType The following choices are available in the list box: <ul style="list-style-type: none"> • Primary (key most frequently used) • Alternate (second most frequently used key) • Candidate (potential, but not preferred key)

Child Relationships

- Identifier connects-to Collection
- Identifier connects-to Column
- Identifier connects-to Key
- Identifier connects-to Table
- Identifier is-composed-of Relationship
- Identifier is-composed-of Attribute
- Identifier is-implemented-by Collection
- Identifier is-implemented-by Column
- Identifier is-implemented-by File
- Identifier is-implemented-by Key
- Identifier is-implemented-by Table

Parent Relationships

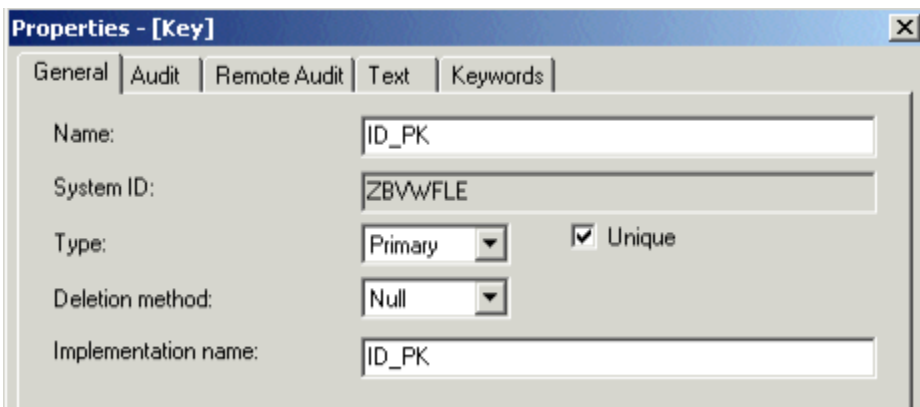
- ER View involves Identifier
- Entity has Identifier

Key

Key

The Key entity type ([Key Properties Dialog](#)) describes a key in a database. You can use a key to enforce referential integrity and uniqueness constraints.

Key Properties Dialog



The types of key are:

- *Foreign*--- A foreign key is a unique index into another table and can be used to join two tables. A foreign key is one or more columns that uniquely identify rows in another table that associates two entities through a relationship.
- *Primary*--- A primary key (the only required key) is one or more unique columns that identifies a single instance (a row in a table) of an entity.
- *Index*--- An index key is one or more non-unique columns that can locate more than one of an entity's instances.

Key Properties		
Field in Dialog	Property	Description
Type	Key_DBType	Status: Required Default is Index mDomDBtypeKey The type of the key. The following choices are available in the list box: <ul style="list-style-type: none">• Foreign• Primary• Index

Unique (check box)	Unique	Status: Required mDomBoolean Specifies whether the key is unique. The choices available are: <ul style="list-style-type: none"> • Yes • No
Deletion method	Key_Delete_Rule	Status: Required Default is Null mDomKeyDeleteRule Indicates the method of the rule deletion. The following choices are available in the list box: <ul style="list-style-type: none"> • Restrict • Cascade • Null
Implementation name	Key_Imp_Name	Status: Optional Maximum of thirty characters
N/A	Update_Rule	Status: Optional

Child Relationships

Key has Column

Parent Relationships

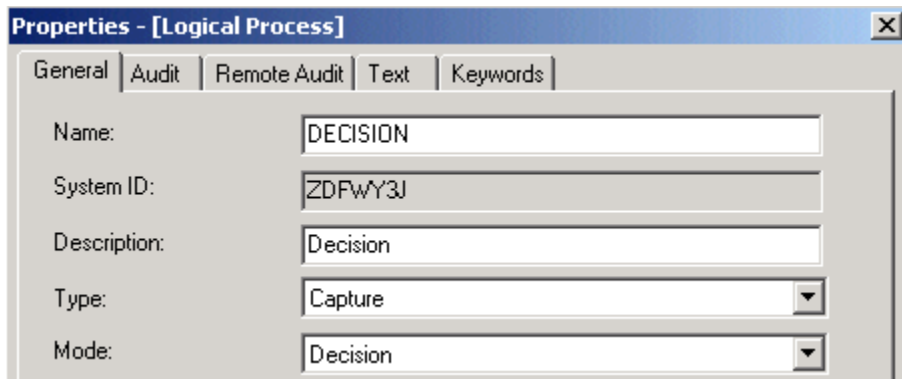
- Attribute connects-to Key
- Attribute is-implemented-by Key
- Collection connects-to Key
- Collection is-implemented-by Key
- Data type connects-to Key
- Data type is-implemented-by Key
- Entity connects-to Key
- Entity is-implemented-by Key
- Identifier connects-to Key
- Identifier is-implemented-by Key
- Relationship connects-to Key
- Relationship is-implemented-by Key
- Table is-implemented-by Key
- Table has Key
- Table is-referenced-by Key

Logical Process

Logical Process

The Logical Process entity type ([Logical Process Properties Dialog](#)) represents an action performed to satisfy a business requirement. A logical process is the smallest unit of work that it is useful to describe.

Logical Process Properties Dialog



Logical Process Properties	
Property	Description
LogPro_Type	<p>Status: Required Default is Capture mDomLogicalProcessType Designates the action the Logical Process performs. The following choices are available in the list box:</p> <ul style="list-style-type: none"> Capture Update Delete Associate Retrieve Calculate Validate
LogPro_Mode	<p>Status: Required Default is Action mDomLogicalProcessMode Designates the type of Logical Process. The following choices are available in the list box:</p> <ul style="list-style-type: none"> • Decision • Action • Logical function
ResponseNeeded	<p>Status: Optional Default is Day mDomDuration The following choices are available in the list box:</p> <ul style="list-style-type: none"> • Day • Week • Month • Year • Quarter • Minute • Hour • Second
TransationCenter	<p>Status: Optional Default is No mDomBoolean The following choices are available in the list box:</p> <ul style="list-style-type: none"> • Yes • No
CentralTransform	<p>Status: Optional Default is No mDomBoolean The following choices are available in the list box:</p> <ul style="list-style-type: none"> • Yes • No
LeafProcess	<p>Status: Optional Default is No mDomBoolean The following choices are available in the list box:</p> <ul style="list-style-type: none"> • Yes • No
Frequency	<p>Status: Optional Can contain a maximum of four characters Default is zero</p>

FrequencyUnit	Status: Optional Default is Day mDomDuration The following choices are available in the list box: <ul style="list-style-type: none"> • Day • Week • Month • Year • Quarter • Minute • Hour • Second
---------------	--

Child Relationships

- Logical Process affects Entity
- Logical Process contains Data Flow
- Logical Process data-content-defined-by ER View
- Logical Process depends-on Logical Process
- Logical Process has-process-dependency Drawing
- Logical Process initiates Event
- Logical Process is-composed-of Logical Process
- Logical Process is-implemented-by Rule
- Logical Process is-signed-by Data Flow
- Logical Process is-supported by Opportunity
- Logical Process is-supported by Problem
- Logical Process supports Information Need
- Logical Process supports Success Factor

Parent Relationships

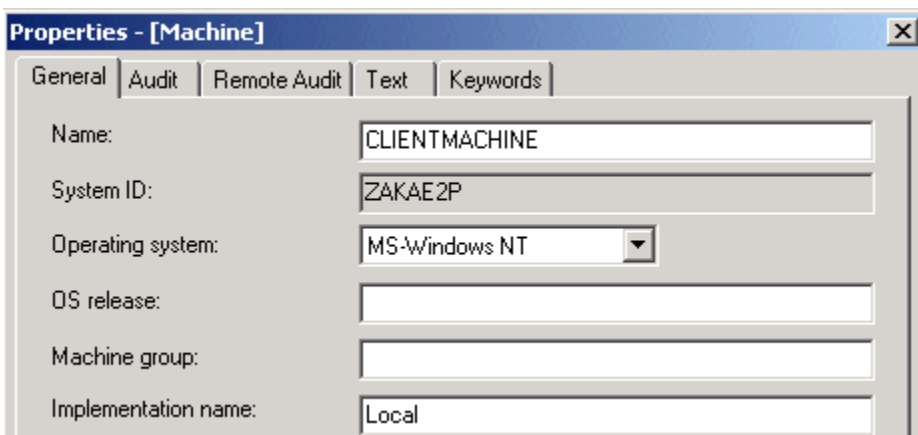
- Assumption is-supported-by Logical Process
- Business Object owns Logical Process
- Context Diagram contains Logical Process
- Development Project includes Logical Process
- Entity is-modified-by Logical Process
- Event triggers Logical Process
- Goal is-supported by Logical Process
- Problem affects Logical Process
- Problem is-caused by Logical Process
- Transition triggers Logical Process

Machine

Machine

The Machine entity type ([Machine Properties Dialog](#)) represents a workstation in a physical network.

Machine Properties Dialog



The screenshot shows a dialog box titled "Properties - [Machine]". It has five tabs: "General", "Audit", "Remote Audit", "Text", and "Keywords". The "General" tab is selected. The fields are as follows:

- Name: CLIENTMACHINE
- System ID: ZAKAE2P
- Operating system: MS-Windows NT (dropdown menu)
- OS release: (empty text box)
- Machine group: (empty text box)
- Implementation name: Local

Machine Properties		
Field in Dialog	Property	Description
Operating system	OS_Type	Status: Required Default is MS-Windows mDomOSType The operating system executing on the machine. The following choices are available in the list box: <ul style="list-style-type: none"> • MS-Windows • MVS • AIX • SunOS • HPUX
OS release	OS_Release	Status: Optional The version of the operating system executing on the machine. Maximum of thirty characters.
Implementation name	Mc_Imp_Name	Status: Optional Can contain a maximum of thirty characters Default is Local When specifying the TCP/IP host name in a machine object, you must specify the full domain name. However, dots (periods) are invalid characters in the implementation name property. Therefore, you must specify the machine with its IP address in the hosts file.

Child Relationships

Machine can-access Machine

Parent Relationships

- Cell contains Machine
- Partition encapsulates Machine

Migration

Migration

The Migration entity type ([Migration Properties](#)) is the focus of all migration activity. You use it to find information about migration, to specify migration action processing options, to view the results of batch processing jobs, and to organize and manage routine migrations. It can be thought of as the envelope that contains the object, object hierarchies, or migration root entity during source repository processing.

Migration Properties	
Property	Description
Mig_State	Status: Optional Can contain a maximum of four characters Default is No Action Executed mDomMigState Refer to Migration states for a list of migration states and their values.
Mig_DSN	Status: Optional Can contain a maximum of 44 characters
Return_Code	Status: Optional Can contain a maximum of 4 characters

The values for the migration states are summarized in [Migration states](#).

Migration states	
Description	Values
No Action Executed	0
Export Job Submitted	0001

Export Job Executing	0002
Export Job Failed	0003
Export Job Successful	0004
Export Job Approved	0005
Load Job Submitted	1001
Load Job Executing	1002
Load Job Failed	1003
Load Job Successful	1004
Load Job Approved	1005
Analyze Job Submitted	2002
Analyze Job Executing	2002
Analyze Job Failed	2003
Analyze Job Successful	2004
Analyze Job Approved	2005
Import Job Submitted	3001
Import Job Executing	3002
Import Job Failed	3003
Import Job Successful	3004
Import Job Approved	3005

Child Relationships

Migration is-profiled-by Group

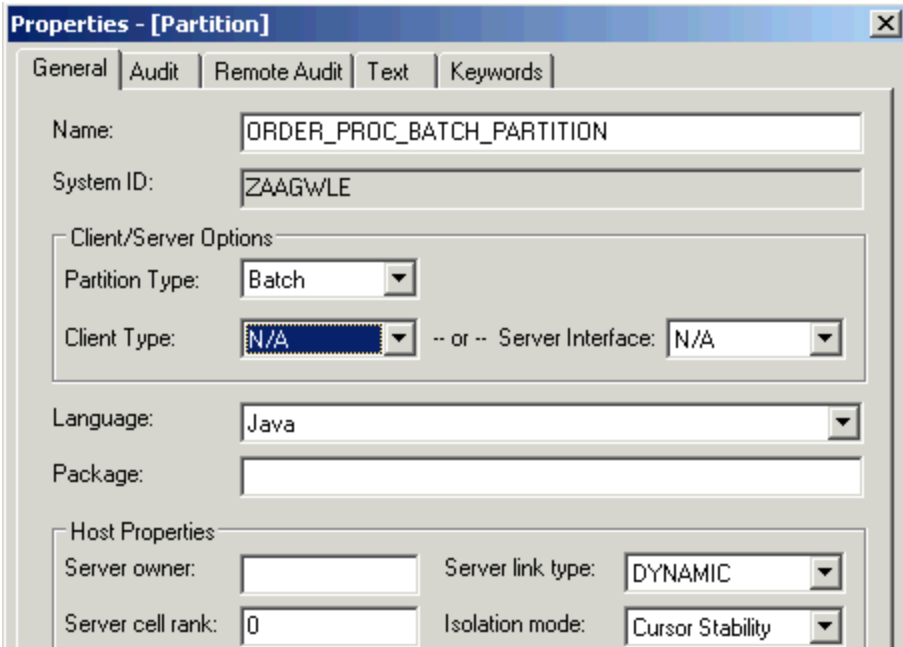
The Migration entity type can also be related to all other named entity types. For example, a Migration entity type can have as its root an Application Configuration entity type or an Attribute entity type or any other named entity type.

Partition

Partition

The Partition entity type ([Partition Properties Dialog](#)) is an association between a client or server process and a machine and its database.

Partition Properties Dialog



The method of linking a rule that belongs to the associated server can be:

- *Dynamic* ---The code needed to resolve external references is shared with other rules and so any external references are resolved at runtime.
- *Static* ---The code needed to resolve external references is copied into the rule's executable load module at link time so any external references are resolved at prepare time.

Partition Properties	
Property	Description
GeneratedLanguage	Status: Required Java is the only choice available mDomGeneratedLanguage
ImplementationPackage	Status: Required Maximum of thirty characters.
ServerInterface	Status: Required Default is mDomServerInterface The network protocol on which the associated Server listens can be:

- EJB
- MQSeries
- N/A
- NetEssential
- RMI
- WebService|

Srv_LinkType	Status: Required Default is Dynamic mDomSrvLinkType The method of linking a rule that belongs to the associated server can be:
--------------	---

- *Dynamic*
- *Static* |

PartitionType	Status: Required Default is Client mDomPartitionType The method of starting the associated Server can be:
---------------	--

- Batch

- Client
- Server
- Gateway|

Srv_CellRank	Status: Optional This can be a numeric Value from 0 to 9.
Srv_Owner	Status: Optional Can contain a maximum of 8 characters The identifier according to which a DB2 plan or package is prepared for all rules that belong to the associated server. Maximum of thirty characters.
Srv_Qualifier	Status: Optional The DB2 identifier for all unqualified host SQL calls associated with the configuration unit. Maximum of thirty characters.
Srv_PlanName	Status: Optional The DB2 plan for the associated rules. Maximum of thirty characters.
CollectionId	Status: Optional The DB2 collection ID to be used for objects that are connected to this configuration unit. Maximum of thirty characters.
IsolationMode	Status: Required Default is Cursor Stability mDomIsolationMode The isolation mode to be used for objects that are connected to this configuration unit. The following choices are platform specific and are dependent on DB2 settings:

- Committed Read
- Cursor Stability
- Dirty Read
- Repeatable Read
- Not Applicable|

MinTranId	Status: Optional Can contain a maximum of eight characters The starting trans ID to be used for objects that are connected to this configuration unit.
MaxTranId	Status: Optional Can contain a maximum of eight characters The ending trans ID to be used for objects that are connected to this configuration unit.
ClientType	Status: Required Default is Event Driven mDomClientType The following choices are available:

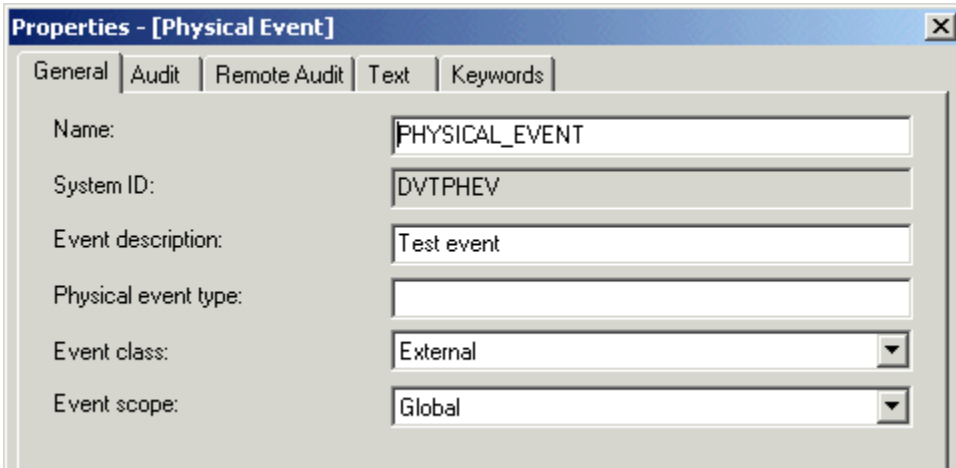
- HTML
- Event Driven
- Converse
- N/A| **Child Relationships**
- Partition encapsulates Component
- Partition encapsulates Database
- Partition encapsulates File
- Partition encapsulates Function
- Partition encapsulates Machine
- Partition encapsulates Process
- Partition encapsulates Report
- Partition encapsulates Rule
- Partition encapsulates Server
- Partition encapsulates Set
- Partition encapsulates Window
- Partition is-had-by Application Configuration
- Partition uses Language

Physical Event

Physical Event

The Physical Event entity type ([Physical Event Properties Dialog](#)) represents a user-defined (business) event. Examples of such an event could be a change in the price of a stock or the withdrawal of money greater than a specified amount.

Physical Event Properties Dialog



Properties - [Physical Event]

General | Audit | Remote Audit | Text | Keywords

Name: PHYSICAL_EVENT

System ID: DVTPEV

Event description: Test event

Physical event type:

Event class: External

Event scope: Global

Physical Event properties	
Property	Description
Event_Description	Status: Optional Up to thirty characters of additional information that describes the event.
PhyEvent_Type	Status: Optional Can contain a maximum of 32 characters
Event_Class	Status: Required Can contain a maximum of two characters Default is External mDomEventClass The following choices are available: <ul style="list-style-type: none">• External• Internal• Temporal
Event_Scope	Status: Required Default is Global mDomEventScope The following choices are available: <ul style="list-style-type: none">• Local• Global• Cell

Child Relationships

- Physical Event has Rule
- Physical Event owns View

Parent Relationships

Rule triggers Physical Event

Process

Process

The Process entity type ([Process Properties Dialog](#)) is an activity that comprises logical units of work (LUWs). Each process represents a single application (leaf process) or a set of applications. The hierarchy of processes you create in your repository is transformed in the application execution environment into an end-user menu.

Process Properties Dialog

Properties - [Process]

General | Audit | Remote Audit | Text | Keywords

Name: PROC`

System ID: ZAJZU2P

Menu description:

Sub-process menu type: Menu Bar

Description: Menu Bar
No Menu (Batch)
Pull Down

Workstation group:

Execution environment: PC Workstation

Process Properties	
Property	Description
Proc_Name	Status: Optional Can contain a maximum of eighteen characters
Menu_Desc	Status: Required Up to thirty characters of text for the menu choice the user chooses to invoke the process.
Work_Station_Group	Status: Optional Can contain a maximum of 8 characters
Child_Menu	Status: Required Default is Menu Bar mDomChildMenu Indicates whether the processes under this function appear on a pull-down menu or on a new menu bar at the top of a window. The following choices are available in the list box: <ul style="list-style-type: none"> • Menu Bar • Pull-Down • Hidden (Batch)
Process_Execenv	Status: Generated Can contain a maximum of eight characters Default is PC Workstation mDomChildMenu This property appears only after a Super Prepare or a Setup action has been performed. This property may be one of the following values: <ul style="list-style-type: none"> • Not Applicable • IBM Mainframe (CICS) • PC Workstation • PC & IBM Mainframe (CICS) • PC & IBM Mainframe (IMS) • IBM Mainframe (IMS)

Child Relationships

- Process depends-on Process
- Process has Bitmap
- Process impacts Entity
- Process is-defined-by Rule
- Process refines-into Process

Parent Relationships

- Partition encapsulates Process
- Event triggers Process
- Function refines-into Process
- Process is-dependent-on Process
- Process is-refined-from Process

Relationships

Relationship

The Relationship entity type ([Relationship Properties Dialog](#)) records information about the relationship type between two or more entities, between an entity and another relationship, or between two or more relationships. Typically, you define instances of relationship entity types during business object analysis when you build your data model. For example, if you have two entity instances, Reservation and Car Type, you might have an instance of a relationship called Specifies to describe how they are connected.

Relationship Properties Dialog

Relationship Properties	
Property	Description
Cmpx_Indicator	<p>Status: Required Default is Regular mDomComplexIndicator The following values are available:</p> <ul style="list-style-type: none"> • Regular • AND • XOR • IOR • Subtype <p>You can change the value of this property only by changing the type of the relationship in the Entity-Relationship Diagrammer. The values AND, IOR, and XOR that appear in the enterprise repository are reserved for future use.</p>

Exp_Min_Rows	Status: Optional Can be a maximum of five characters. User-defined lower limit of the expected size of the instance of the entity type (number of Fields). This can be a numeric value from 0 to 99,999.
Exp_Max_Rows	Status: Optional Can be a maximum of ten characters. User-defined upper limit of the expected size of the instance of the entity type (number of fields). This can be a numeric value from 0 to 9,999,999,999.
Act_Period	Status: Required Can contain a maximum of five characters Default is Day mDomDuration The following values are available: <ul style="list-style-type: none"> • Day • Week • Month • Year • Quarter • Minute • Hour • Second
Ave_Wkly_Inserts	Status: Optional Can contain a maximum of 4 characters Estimated number of times during implementation that the instance of this entity type will be created. This can be a numeric value from 0 to 9,999.
Ave_Wkly_Deletes	Status: Optional Can contain a maximum of 4 characters Estimated number of times during implementation that the instance of this entity type will be deleted. This can be a numeric value from 0 to 9,999.
Ave_Wkly_Updates	Status: Optional Can contain a maximum of 4 characters Estimated number of times during implementation that the instance of this entity type will be updated. This can be a numeric value from 0 to 9,999.
Exp_Rows	Status: Optional Can contain a maximum of 10 characters The number of instances of this entity expected when you define the data model. This can be a numeric value from 0 to 9,999,999,999.

Child Relationships

- Relationship connects-to Collection
- Relationship connects-to Column
- Relationship connects-to Key
- Relationship connects-to Table
- Relationship is-described-by Attribute
- Relationship is-implemented-by Collection
- Relationship is-implemented-by Column
- Relationship is-implemented-by File
- Relationship is-implemented-by Key
- Relationship is-implemented-by Table
- Relationship relates-via Relationship

Parent Relationships

- Attribute is-composed-of Relationship
- Identifier is-composed-of Relationship
- Entity is-related-via Relationship
- ER View involves Relationship
- Relationship is-related-via Relationship

Report

Report

The Report entity type ([Report Properties Dialog](#)), in conjunction with the section entity type, defines the paper-based output that an application produces for the end user. This application is usually a batch application. You typically define an instance of a report entity type using the Report Painter tool in the Construction Workbench.

Report Properties Dialog

The screenshot shows a dialog box titled "Properties - [Report]". It has five tabs: "General", "Audit", "Remote Audit", "Text", and "Keywords". The "General" tab is active. The following fields are visible:

- Name: ALLREPORT
- System ID: ZAKUY2P
- Execution environment: IBM Mainframe Batch (MVS)
- Page size: 0060
- Line size: 0072
- Left margin: 0000
- Top margin: 0000
- Printer type: Generic
- Orientation: Portrait
- Implementation name: ZAKUY2P

Report Properties	
Property	Description
Exec_Environ	<p>Status: Required Can contain a maximum of six characters Default is MVSBAT mDomReportExecEnv The following values are available:</p> <ul style="list-style-type: none"> • IBM Mainframe Batch • IBM Mainframe (CICS) • Workstation PM
Page_Size	<p>Status: Required Can contain a maximum of four characters Default is 60 Number of lines or inches (depending on the printer type) per page for the report. This can be a numeric value from 1 to 500.</p>
Line_Size	<p>Status: Required Can contain a maximum of 4 characters Default is 72 Width in characters or inches (depending on the printer type) per line for the report. This can be a numeric value from 1 to 132.</p>
Left_Margin	<p>Status: Required Can contain a maximum of four characters Column number where printing begins. This can be a numeric value from 0 to the line size.</p>
Top_Margin	<p>Status: Required Can contain a maximum of four characters Row number where printing begins. This can be a numeric value from 0 to the page size.</p>

PrinterType	Status: Required Default is Generic mDomPrinterType The following choices are available in the list box: <ul style="list-style-type: none"> • Generic (character specification) • 38000 (inches/10 specification) Page size and line size width are interpreted as tenths of an inch for 3800 printers and the number of characters for generic printers.
Orientation	Status: Required Default is Portrait mDomOrientation The following choices are available in the list box: <ul style="list-style-type: none"> • Portrait • Landscape
Report_Imp_Name	Status: Optional Maximum of 7 alphanumeric characters, the first of which must be alphabetic.

Child Relationships

- Report contains Section
- Report has Bitmap
- Report has Form
- Report owns View
- Report refers-to Set

Parent Relationships

- Partition encapsulates Report
- Rule converses Report

Rule

Rule

The Rule entity type ([Rule Properties Dialog](#)) records reusable procedural system specifications for your application. Rules define the logic of a process, control the execution of other rules and components, invoke windows and reports, and access files. The logic in a rule is specified using the Rules Language.

Ideally, each rule represents one logical unit of work. That is, each rule should accomplish one task only. This approach maximizes the potential for reuse. If you combine several tasks into one rule and then perform one or two of them again, it may not be feasible to reuse your original rule because it also performs unneeded tasks.

Rule Properties Dialog

Properties - [Rule]

General | Audit | Remote Audit | Text | Keywords

Name:

System ID:

Execution Environment:

DBMS Usage:

Execution Mode:

Plan Name:

Description:

Implementation Name:

Package:

Rule Properties	
Property	Description
Exec_Environ	<p>Status: Required Can contain a maximum of five characters Default is PC Workstation mDomRuleExecEnv The following choices are available in the list box:</p> <ul style="list-style-type: none"> • IBM Mainframe (CICS & Batch) • PC Workstation • IBM Mainframe (IMS) • IBM Mainframe (CICS) • IBM Mainframe Batch • PC & IBM Mainframe (CICS) • PC & IBM Mainframe (IMS)
Rule_Source	<p>Status: Required Can contain a maximum of eight characters</p>
Sys_Source	<p>Status: Required Can contain a maximum of three characters Default is Not Applicable mDomRuleDBMSUsage Defines whether the rule accesses a database table. The following choices are available in the list box:</p> <ul style="list-style-type: none"> • DB2 • Not Applicable <p>Note: Use the DB2 value whether the database is DB2, DBM, Informix or other databases.</p>
Rule_Name	<p>Status: Required Can contain a maximum of thirty characters</p>
Rule_Str_Id	<p>Status: Required Can contain a maximum of five characters Default is Synchronous execution mDomExecuteMode Defines the type of processing during execution. The following choices are available in the list box:</p> <ul style="list-style-type: none"> • Synchronous execution • Asynchronous execution • Event Driven execution
Rule_TranId	<p>Status: Optional Can contain a maximum of eight characters</p>

Rule_Imp_Name	Status: Optional Maximum of 8 alphanumeric characters, the first of which must be alphabetic.
Package	Status: Optional The name of the LANDP Server that the rule accesses. Maximum of eight alphanumeric characters. Any character but the first can also be a number sign (#).
Isolation_Mode	Status: Required Default is Committed Read mDomIsolationMode The following choices are available in the list box: <ul style="list-style-type: none"> • Committed Read • Cursor Stability • Dirty Read • Repeatable Read • Not Applicable

Child Relationships

- Rule converses Report
- Rule converses Window
- Rule depends-on Rule
- Rule has Bitmap
- Rule is-accessed-by File
- Rule owns View
- Rule refers-to Set
- Rule triggers Physical Event
- Rule uses Component
- Rule uses Rule

Parent Relationships

- Partition encapsulates Rule
- Logical Process is-implemented-by Rule
- Physical Event has Rule
- Process is-defined-by Rule
- Rule is-dependend-on-by Rule
- Rule is-used-by Rule
- Server contains Rule

Section

Section

The Section entity type ([Section Properties Dialog](#)), in conjunction with the Report entity type, defines the paper-based output that an application produces for an end user. The application is usually a batch application. Each section includes definition information about a particular part of the report, such as a header section or a footer section. You typically create a report and its associated sections with the Report Painter tool.

A section type is defined on the *Report > Section relationship properties panel*, not on the *Section object properties panel*.

Section Properties Dialog

Section Properties	
Property	Description

ShortName	Status: Optional Can contain a maximum of 8 characters
Section_Imp_Name	Status: Optional Maximum of 8 alphanumeric characters, the first of which must be alphabetic

Child Relationships

- Section owns View
- Section uses Language

Parent Relationships

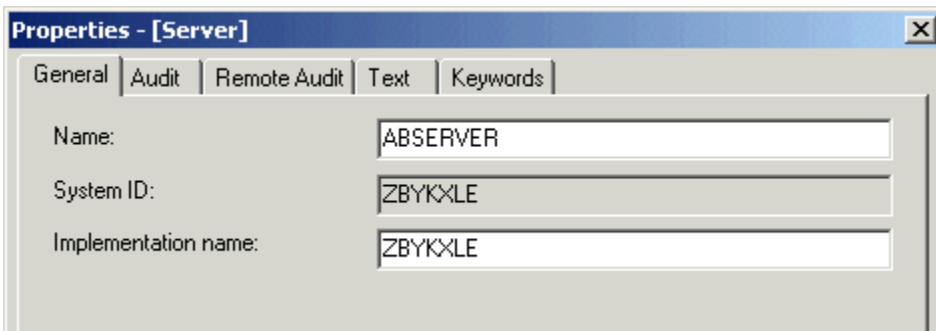
Report contains Section

Server

Server

The Server entity type represents a Server Process in a network configuration.

Server Properties Dialog



The screenshot shows a dialog box titled "Properties - [Server]". It has five tabs: "General", "Audit", "Remote Audit", "Text", and "Keywords". The "General" tab is active. It contains three text input fields: "Name" with the value "ABSERVER", "System ID" with the value "ZBYKXLE", and "Implementation name" with the value "ZBYKXLE".

Server Properties	
Property	Description
Srv_Imp_Name	Status: Optional Can contain a maximum of 30 characters

Child Relationships

Server contains Rule

Parent Relationships

Partition encapsulates Server

Set

Set

The Set entity type ([Set Properties Dialog](#)) provides a way to create a named value in the repository that can be referred to by any number of rules in any number of applications. Sets can be used in the following ways:

System Choices

This use of a set allows choices from a combination box in an application's window. For example, a user might have to choose the type of car: Compact, Sub-Compact, or Full-Size.

Messages

Predefined messages can be defined in a set referred to by the standard component:

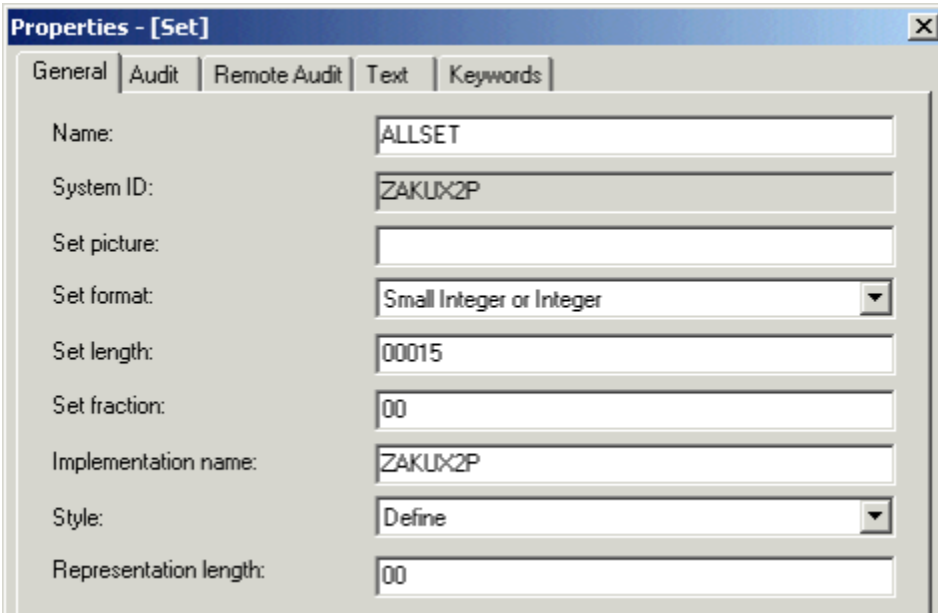
SET_WINDOW_MESSAGE

to display an error:

Symbolic name Sets

Messages are used primarily to make Rules Language code more readable by allowing developers to substitute meaningful references for cryptic variables. For example, when you create a message set, each message is associated with a value (such as 100). To invoke a particular message you can specify that value in your Rules Language code. Alternately, you can associate a meaningful symbolic name with the message (for example, RECORD_NOT_FOUND) and refer to the message using the symbolic name instead of the value. For more information about sets, see [Contains \(is-contained-in\)](#).

Set Properties Dialog



Set Properties		
Field in Dialog	Property	Description
Set format	Set_Type	Status: Required Default is Small Integer or Integer mDomSetType Data type of the Set values. The following choices appear in the list box: <ul style="list-style-type: none"> • Character • Decimal • Graphic Character (DBCS) • Mixed Character • Small Integer or Integer
Set length	Set_Length	Status: Required Can contain a maximum of 5 characters Default is 0015 The value range for this property depends on the Set Type.
Set fraction	Set_Fraction	Status: Required if Set Format is Decimal Status: Irrelevant for any other Set Format Can contain a maximum of 2 characters Number of decimal places in the set's values. For decimal sets, the set fraction must be greater than 0 and less than or equal to the Set length.
Implementation name	Set_Imp_Name	Status: Optional Can contain a maximum of 8 alphanumeric characters, the first of which must be alphabetic

Style	Set_Style	Status: Required Default is Values mDomSetStyle The following choices are available in the list box: <ul style="list-style-type: none"> • Define • Error • Lookup • Values Refer to Set Styles available in Set Builder for a description of set styles available in Set Builder
Representation length	RepresentationLen	Status: Optional Can contain a maximum of 2 characters whose numeric value must be from 0 to 99

Set Lengths by Set Format			
Set format	Set length		Set fraction
	Minimum length	Maximum length	
Character	1	30	–
Decimal	1	31	No more than Set length.
Graphic character (DBCS)	1	15	–
Mixed character	1	30	–
Small integer or integer	Small: 15; Integer: 31	Small: 15; Integer: 31	–

Set Styles available in Set Builder	
Style	Description
Define	For restrictions on Define and Encoding properties, refer to Symbol . Define and Encoding properties are required and Define must be unique.
Error	The Define, Encoding, and Display fields are required and must have unique values. The Set type can only be Small Integer.
Lookup	The Define, Encoding, and Display fields are required and must have unique values.

Child Relationships

- Set contains Symbol
- Set contains Value

Parent Relationships

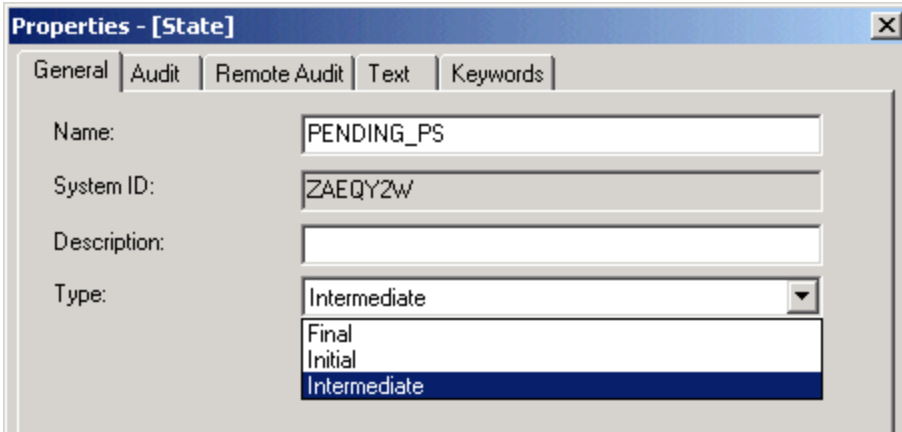
- Component refers-to Set
- Configuration unit encapsulates Set
- Data type is-constrained-by Set
- Field refers-to Set
- Report refers-to Set
- Rule refers-to set Set
- Window refers-to set Set

State

State

The State entity type ([State Properties Dialog](#)) is a discrete set of properties, values, and relationships a data object holds at a point in time.

State Properties Dialog



State Properties	
Property	Description
Description	Status: Optional Up to thirty characters of additional information that describes the state
State_Type	Status: Required Default is Intermediate mDomStateType Designates the type of state. The following choices are available in the list box: <ul style="list-style-type: none"> • <i>Initial</i>--- An initial state represents the condition of a data object before any processing and transitions • <i>Intermediate</i>--- An intermediate state is a state a data object resides in during most of its life cycle, and which it leaves or enters only when some external or internal event triggers a state change • <i>Final</i>--- A final state is a state the data object enters and does not leave

Child Relationships

State has-a-state-transition diagram Drawing

Parent Relationships

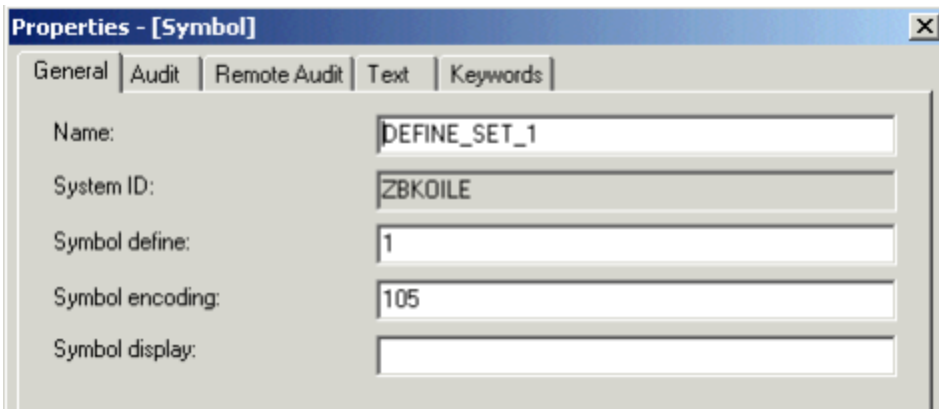
- Business Object has State
- Entity has State
- Transition is-preconditioned-by State
- Transition results-in State

Symbol

Symbol

The Symbol entity type ([Symbol Properties Dialog](#)) is used for the items in a set. The properties dialog is available using the Repository Administration Tool.

Symbol Properties Dialog



Symbol properties	
Property	Description
Define	Status: Optional Can contain a a maximum of eight characters The name that is used in a rule to refer to the symbol. (Example: red in SET_OF_COLORS) Should start with a letter or underscore (_) followed by a sequence of letters, digits, or underscores.
Encoding	Status: Optional Can contain a a maximum of 8 characters The value of the symbol that is stored in fields. (Example: 1)
Display	Status: Optional Can contain a a maximum of eight characters The display value of the symbol used for lookup sets. (Example: Red)

Child Relationships

Symbol uses Language

Parent Relationships

Set contains Symbol

Table

Table

The Table entity type ([Table Properties Dialog](#)) describes a table or view in a database. Its properties store volumetric and implementation data.

Table Properties Dialog

Properties - [Table]

General | Audit | Remote Audit | Text | Keywords

Name: LOCATION

System ID: ZBVWOLE

Table or view: Table

Table creator:

Expected minimum rows: 00000

Expected maximum rows: 0000000000

Activity period: Day

Avg. inserts for period: 0000

Avg. deletes for period: 0000

Avg. updates for period: 0000

Implementation name: LOCATION

Table Properties	
Property	Description
Table_View	<p>Status: Required Default is Table mDomTableView The following choices are available in the list box:</p> <ul style="list-style-type: none"> • Table • View
Tbl_Imp_Name	<p>Status: Required Can contain a maximum of 30 characters</p>
Exp_Min_Rows	<p>Status: Optional Can contain a maximum of 5 characters User-defined lower limit of the expected size of the instance (number of fields). This can be a numeric value from 0 to 99999 that is less than the expected maximum rows property.</p>
Exp_Max_Rows	<p>Status: Optional Can contain a maximum of 10 characters User-defined upper limit of the expected size of the instance (number of fields). This can be a numeric value from 0 to 9999999999.</p>
Act_Period_s	<p>Status: Required Default is Day mDomDuration_s Time period that describes the frequency of various operations performed with instances of this entity type. The following choices are available in the list box:</p> <ul style="list-style-type: none"> • Day • Week • Month • Year • Quarter
Ave_Inserts	<p>Status: Optional Can contain a maximum of four characters Estimated number of times during implementation that instances of this entity type will be created. This can be a from 0 to 9999.</p>

Ave_Deletes	Status: Optional Can contain a maximum of four characters Estimated number of times during implementation that instances of this entity type will be deleted. This can be a numeric value from 0 to 9999.
Ave_Updates	Status: Optional Can contain a maximum of four characters Estimated number of times during implementation that instances of this entity type will be updated. This can be a numeric value from 0 to 9999.
Creator	Status: Optional Can contain a maximum of eight characters

Child Relationships

- Table has Column
- Table has Key
- Table implements Table
- Table is-based-on Table
- Table is-implemented-by Collection
- Table is-implemented-by Column
- Table is-implemented-by File
- Table is-implemented-by Key
- Table is-implemented-by Table
- Table is-referenced-by Key

Parent Relationships

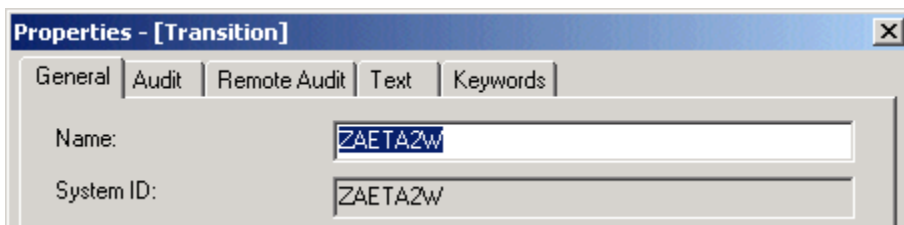
- Attribute connects-to Table
- Attribute is-implemented-by Table
- Collection connects-to Table
- Collection is-implemented-by Table
- Data type connects-to Table
- Data type is-implemented-by Table
- Database contains Table
- Entity connects-to Table
- Entity is-implemented-by Table
- Identifier connects-to Table
- Identifier is-implemented-by Table
- Relationship connects-to Table
- Relationship is-implemented-by Table

Transition

Transition

The Transition entity type ([Transition Properties Dialog](#)) represents an event and the processes it triggers that cause an entity to move from one state to another.

Transition Properties Dialog



Child Relationships

- Transition data-content-defined-by ER View
- Transition is-caused-by Event
- Transition is-preconditioned-by State
- Transition results-in State
- Transition triggers Logical Process

Parent Relationships

Event causes Transition

Transition entity type	
Property	Description
Name	Status: Required Common Property
System ID	Status: Generated Common Property
ShortName	Status: Optional Can contain a maximum of 8 characters

Value

Value

The Value entity type ([Value Properties Dialog](#)) represents the Value entity.

Value Properties Dialog

Value Properties		
Field in Dialog	Property	Description
Value	Description	Status: Optional Up to thirty characters of additional information that describes the Value.

Child Relationships

Value is-classified-by Scheme (Implemented as: Value classifies Scheme)

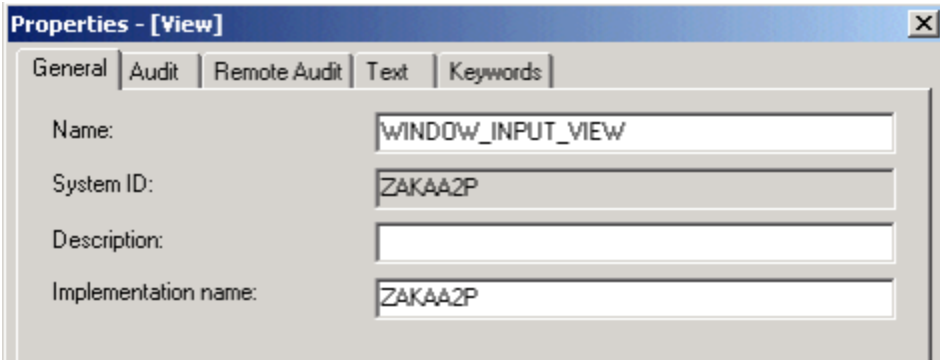
View

View

The View entity type ([View Properties Dialog](#)) defines data structures in the AppBuilder environment. For example, a view defines the input to or the output from a rule, or the fields that can appear on a window. Each view can be thought of as a node in a data tree, the leaf nodes of which are fields. The root view of each such structure is associated with a rule, a component, a window, a file, or a section entity type through an *owns* relationship type, and with fields and other views through an *includes* relationship type. Any included views can include other views and fields. Each view in a data tree can be considered the root node of a subtree. In the AppBuilder environment, each of these subtrees containing a view, and the views and fields that descend from it, is called a *view structure*. Various configurations of view structures make up all AppBuilder environment data structures.

Through their views, rules can accept parameters, return data, and communicate with the end user through windows. For example, a rule can place data in the input view of a rule it uses, call the rule, and, when the called rule terminates, read the returned data from its output view. A rule can also place data into the fields of the window view of a window it converses, then call the window to display the data to the end user. When the end user closes the window, the rule can read the fields the end user changed.

View Properties Dialog



View Properties	
Property	Description
View_Name	Status: Optional Can be a maximum of 18 characters of additional information that describes the view
View_Imp_Name	Status: Optional Can be a maximum of 8 alphanumeric characters, the first of which must be alphabetic

Child Relationships

- View includes Field
- View includes View

Parent Relationships

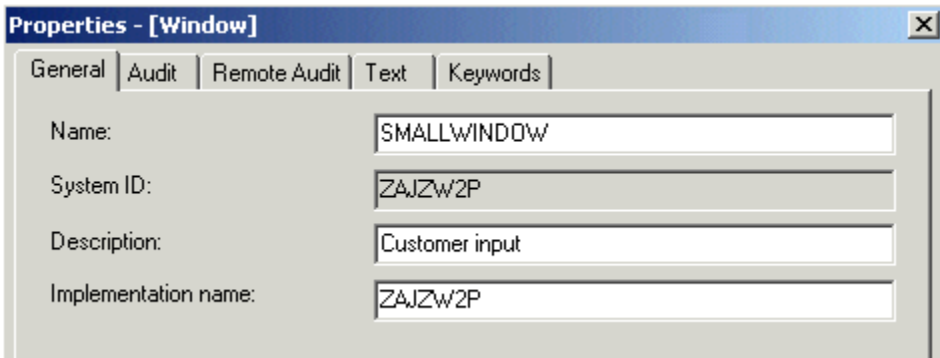
- View is-included-in View
- Component owns View
- Entity owns View
- File owns View
- Physical Event owns View
- Report owns View
- Rule owns View
- Section owns View
- Window owns View

Window

Window

The Window entity type ([Window Properties Dialog](#)) is the logical representation in a repository of a window the user sees on the workstation screen or a 3270 panel. To create the window, use the Window Painter tool of the Construction Workbench. The description of your panel is then stored in the repository as a window entity type. Each window is associated with one view, called its window view, through the owns relationship type. This view stores input to and output (new or modified data for the end user) from the window.

Window Properties Dialog



Window Properties	

Property	Description
Window_Name	Status: Optional Can be a maximum of 18 characters of additional information that describes the view
Window_Imp_Name	Status: Optional Can be a maximum of 8 alphanumeric characters, the first of which must be alphabetic

Child Relationships

- Window has Bitmap
- Window has Panel
- Window has Help
- Window has Help Text
- Window has Window Content
- Window owns View
- Window refers-to Set

Parent Relationships

- Partition encapsulates Window
- Rule converses Window

Relationship Types

Relationship Types

A relationship type is used to describe different forms of association between entity types. It can be thought of as the verb that connects the two entities. A relationship type takes its name from this definition. The name is a concatenation of:

subject entity type + relationship type + object entity type

For example, the relationship name for a *uses* relationship type between two rules is Rule uses Rule.

Relationship types (relationships between entity types) are object types in the repository. The vast majority of relationship types are fixed. They relate specific entity types in specific ways that seldom change. For the most part, these relationship types do not have properties associated with them, aside from a sequence number (for example, so you can tell which field has precedence in a data structure).

This topic includes:

- [Common Properties of Relationship Types](#)
- [Reference of Relationship Types](#)

There are two ways to view a relationship type: from the point of view of the parent entity type *or* from the point of view of the child entity type. That is, given a converse relationship between a rule and a window, the Rule converses the Window, and the Window *is-conversed-by* the Rule. The use of the parent name to child name relationship perspective is preferred.

In [Reference of Relationship Types](#), the relationship types are listed in alphabetical order by the preferred name; proper terminology for the secondary name-is shown in parentheses.

Common Properties of Relationship Types

Common Properties of Relationship Types

All relationship types have common properties. All common properties, except parent name, child name, sequence number, and separator identifier are the audit properties of the relationship type. The audit properties (except for local date and local time) are not populated until the object type has been uploaded to the host.

In the personal repository on a workstation, parent name, child name, sequence number, separator identifier, and all properties unique to each relationship type appear on the General tab of the Properties dialog for that relationship type. Audit properties appear on the Object Audit Information window of the relationship type (see [Audit and Remote Audit Tabs](#)).

Properties are displayed differently in the enterprise repository on the host. *All* properties appear on the main screen for each relationship type. Parent name is the first entity type listed on the screen, and child name is the second entity type listed. In the personal repository, parent and child names are generated from the position of objects in drawings. In the enterprise repository, enter these names to establish the relationship type. Property names are slightly different in the three repositories, and some properties do not appear in all. The following list indicates how each property appears in the repositories. If not specified, the name is the same in all repositories.

The common properties of relationship types in the AppBuilder Information Model include the following:

- [Parent Name](#)
- [Child Name](#)
- [Sequence Number \(Personal or Workgroup only\)](#)
- [Separator Identifier \(Personal or Workgroup only\)](#)
- [Internal Name \(visible in Personal or Workgroup only\)](#)

- [ChangeNumber](#)
- [Created/Remote date \(Personal or Workgroup\)](#)
- [Date created \(Enterprise\)](#)
- [Created/Remote time \(Personal or Workgroup\)](#)
- [Time created \(Enterprise\)](#)
- [Created/Remote user \(Personal or Workgroup\)](#)
- [Created by \(Enterprise\)](#)
- [Maintained/Remote date \(Personal or Workgroup\)](#)
- [Date maintained \(Enterprise\)](#)
- [Maintained/Remote time \(Personal or Workgroup\)](#)
- [Time Maintained \(Enterprise\)](#)
- [Maintained/Remote user \(Personal or Workgroup\)](#)
- [Maintained by \(Enterprise\)](#)
- [Maintained/Local date \(Personal or Workgroup only\)](#)
- [Maintained/Local time \(Personal or Workgroup only\)](#)
- [Version \(Enterprise only\)](#)

The following table provides the status and a description for each relationship type.

Common properties of relationship types		
Property	Status	Description
Parent Name	<i>Generated for Personal or Workgroup</i> Required for Enterprise	Name of parent object
Child Name	<i>Generated for Personal or Workgroup</i> Required for Enterprise	Name of child object
Sequence Number (Personal or Workgroup only)	Generated	A number automatically assigned to each entity in sequence and used to define the order of relationships attached to a common entity. You can change this number to change the order of the child entities under a parent entity. This value can be a number from 0 to 999.
Separator Identifier (Personal or Workgroup only)	Generated	Uniquely distinguishes duplicate relationships between the same parent and child entities.
Internal Name (visible in Personal or Workgroup only)	Generated	The AppBuilder environment uses this name internally to refer to the relationship type. This name is not listed with the other information for a relationship type in either repository. However, you might see this name if you encounter trouble during an upload or download. In addition, this name appears in Repository Maintenance when you view Unit of Work. This name distinguishes the different variants of a relationship type. Because this name is not a true property, it is <i>italicized</i> in the descriptions that follow.
ChangeNumber	Generated	System-generated number to protect instances of an object from simultaneous changes from two users. An instance can be uploaded only if its change number matches the change number in the enterprise repository.
Created/Remote date (Personal or Workgroup)	Generated	The date that the instance object was created in the Personal or Workgroup Repository, either directly or by uploading.
Date created (Enterprise)	Generated	Format YY/MM/DD Date the instance was first created in the enterprise repository, either directly or by uploading.

Created/Remote time (Personal or Workgroup)	Generated	The time that the instance object was created in the Personal or Workgroup Repository, either directly or by uploading.
Time created (Enterprise)	Generated	Format: HH:MM Time the instance was first created in the enterprise repository, either directly or by uploading.
Created/Remote user (Personal or Workgroup)	Generated	ID of user who created the instance in the Personal or Workgroup Repository, either directly or by uploading.
Created by (Enterprise)	Generated	ID of user who created the instance in the enterprise repository, either directly or by uploading.
Maintained/Remote date (Personal or Workgroup)	Generated	Date the instance was last changed in the Personal or Workgroup Repository, either directly or by uploading.
Date maintained (Enterprise)	Generated	Format: YY/MM/DD Date the instance was last changed in the enterprise repository, either directly or by uploading.
Maintained/Remote time (Personal or Workgroup)	Generated	Time the instance was last changed in the Personal or Workgroup Repository, either directly or by uploading.
Time Maintained (Enterprise)	Generated	Format: HH:MM Time the instance was last changed in the enterprise repository, either directly or by uploading.
Maintained/Remote user (Personal or Workgroup)	Generated	ID of user who changed the instance in the Personal or Workgroup Repository, either directly or by uploading.
Maintained by (Enterprise)	Generated	ID of last user who changed the instance in the enterprise repository, either directly or by uploading.
Maintained/Local date (Personal or Workgroup only)	Generated	Format: YY/MM/DD Date the instance was last changed on the workstation.
Maintained/Local time (Personal or Workgroup only)	Generated	Format: HH:MM Time the instance was last changed on the workstation.
Version (Enterprise only)	Generated	Identifies the logical repository in which this instance resides.

Many relationship types have more than one variant. A relationship type and its properties might vary depending on the entity types that it relates. Even though two variants might have the same English language name, they are different relationship types.

Reference of Relationship Types

Reference of Relationship Types

This section lists and describes the relationship types in the Information Model. All relationship types are part of the Development Information Model unless otherwise noted.

Each entry in this list starts with a description of the relationship type and which entity types it can connect. Next is a description of the properties of the relationship type, including whether a particular property is required, generated, or optional.

The Relationship Types available in the AppBuilder Information Model are listed in alphabetical order in [Relationship Types](#).

Relationship Types	
Accesses (is-accessed-by)	Is-implemented-by (Implements)
Affects (is-affected-by)	Is-involved-in (Involves)
Can-Access (can-be-accessed-by)	Is-keyed-by (Keys)
Causes (is-caused-by)	Is-modified-at (Where-is-modified)
Cites (is-cited-by)	Is-modified-by (Modifies)

Comprises (is-comprised-of)	Is-preconditioned-by (Preconditions)
Connects (is-connected-to)	Is-profiled-by (Profiles)
Contains (is-contained-in)	Is-referenced-by (References)
Converses (is-conversed-by)	Is-related-via (Relates-to)
Depends-on (is-depended-on-by)	Is-responsible-for (is-the-responsibility-of)
Encapsulates (is-encapsulated-by)	Is-site-of (Has-as-its-site)
Has (is-had-by)	Is-typed-by (Types)
Has STD (has-a-state-transition-diagram)	Maps to (is-mapped-to)
Impacts (is-impacted-by)	Member (is-member-of)
Includes (is-included-in)	Owns (is-owned-by)
Influences (is-influenced-by)	Partitioned (partitioned-by)
Initiates (is-initiated-by)	Redefines
Intersects (intersects-with)	Refers-to (is-referred-to-by)
Invokes (is-invoked-by)	Refines-into (is-refined-from)
Is-carried-out-at (Carries Out)	Replaces System (System is-replaced-by)
Is-composed-of (Composes)	Results-in (is-the-result-of)
Is-constrained-by (Constrains)	Stores (provides-storage-for)
Is-defined-by (Defines)	Supports (Is-supported-by)
Is-described-by (Describes)	Triggers (is-triggered-by)
Is-distributed-at (Where-is-distributed)	Uses (is-used-by)
Is-forwarded-to (Is-forwarded-from)	

Properties

Refer to [Common Properties of Relationship Types](#) for properties common to all entity types. Properties specific to each of the listed Relationship Types are shown in the Properties Table under each entity heading. If there is no Properties Table, there are no properties specific to that relationship entity.

Variants

Each relationship type has the specific variants for that type listed. See [Common Properties of Relationship Types](#) for a definition of variants.

Accesses (is-accessed-by)

Accesses ([Accesses Properties Dialog](#)) is a relationship type in which the first entity type gains entry to the second. For example, a rule or a component accesses or gains entry to a file.

Accesses Properties Dialog

Properties - [File is-accessed-by]

General | Audit | Remote Audit | Text

Parent name:

Child name:

Sequence number:

Separator ID:

Component usage:

Accesses Properties	
Property	Description
Parent name	The system ID of an entity that is accessing a child entity
Child name	The system ID of an entity that is being accessed by a parent entity
Sequence number	A number automatically assigned to each entity in sequence and used to define the order of relationships attached to a common entity. You can change this number to change the order of the child entities under a parent entity. This value can be a number from 0 to 999.
Separator ID	Uniquely distinguishes duplicate relationships between the same parent and child entities.
Component usage	Status: Optional Can contain a maximum of 2 characters

Variants

- Component accesses File (implemented as: File is-accessed-by Component)
- Database accesses File (implemented as: File is-accessed-by Database)
- Rule accesses File (implemented as: File is-accessed-by Rule)

Affects (is-affected-by)

Affects is a relationship type in which the first entity type has an influence on or causes a change in the second entity type.

Affects Properties	
Property	Description
Create flag	Status: Optional Specifies whether the instance of the first entity type creates the second
Read flag	Status: Optional Specifies whether the instance of the first entity type reads the second
Update flag	Status: Optional Specifies whether the instance of the first entity type updates the second
Delete flag	Status: Optional Specifies whether the instance of the first entity type deletes the second
Comments	Status: Optional Can contain a maximum of 30 characters

Variants

Logical Process affects Entity variant (implemented as: Entity is-affected-by Logical Process)

Can-Access (can-be-accessed-by)

Can-Access means that the first entity type can gain entry to the second entity type.

Causes (is-caused-by)

Causes is a relationship type in which the first entity type is the reason for or results in the second entity type.

Causes Properties Dialog

Properties - [Event causes]

General | Audit | Remote Audit | Text

event causes

Parent name: ORDER_ITEM

Child name: ZBXJX3J

Sequence number: 0

Separator ID: 0

Description:

Causes has the following properties:

- Status is Optional.
- Can contain a maximum of thirty characters.

Variants

Event causes Transition (implemented as Transition is-caused-by Event)

Cites (is-cited-by)

Cites means that the first entity mentions or brings forward the second entity as support, proof, or illustration.

Comprises (is-comprised-of)

Comprises is a relationship in which the first entity type comprises the second.

Variants

Window is comprised of Window

Connects (is-connected-to)

Connects-to is a relationship type in which the first entity is connected to the second entity.

Connects Properties Dialog

Properties - [connects-to]

General | Audit | Remote Audit | Text

connects-to

Parent name:

Child name:

Sequence number:

Separator ID:

Variants

The Connects (is-connected-to) relationship type has the following variants:

- Attribute connects-to Collection
- Attribute connects-to Column
- Attribute connects-to Key
- Attribute connects-to Table
- Collection connects-to Collection
- Collection connects-to Column
- Collection connects-to Key
- Collection connects-to Table
- Column connects-to Collection
- Data type connects-to Column
- Data Type connects-to Collection
- Data Type connects-to Key
- Data Type connects-to Table
- Entity connects-to Collection
- Entity connects-to Column
- Entity connects-to Key
- Entity connects-to Table
- Identifier connects-to Collection ([Connects Properties Dialog](#))
- Identifier connects-to Column
- Identifier connects-to Key
- Identifier connects-to Table
- Relationship connects-to Collection
- Relationship connects-to Column
- Relationship connects-to Key
- Relationship connects-to Table

In the enterprise repository, you might see this relationship type listed under more entity types than shown here. Disregard any others and do not use them. They are reserved for future use and are not supported.

Do not modify this relationship type. Doing so can damage the tracing information used by the forward engineering process.

Contains (is-contained-in)

Contains ([Contains Properties Dialog](#)) is a relationship type in which the second entity type is a part of the first entity type.

Contains Properties Dialog

Properties - [Report contains Section]

General | Audit | Remote Audit | Text

Parent name: DVT_REPORT_BATCH

Child name: DVT_SECTION_FOOT

Sequence number: 10

Separator ID: 0

Section type: Footer Section

Page placement: Same Page

Print option: Printed

Break sequence number: 0000

Break field:

Break qualifier:

Left margin: 0000

Variants

This relationship type has the following variants:

- Cell contains Cell (implemented as Cell is-contained-in Cell)
- Cell contains Machine (implemented as Machine is-contained-in Cell)
- Database contains Table
- Group contains Group
- Report contains Section
- Server contains Rule
- Set contains Symbol
- Set contains Value

Report contains Section Properties	
Property	Description
Section type	Status: Required Indicates which type of section appears in the report. The following choices are available in the list box:

- Break section
- Footer section
- Header section
- Regular section

Page placement	Status: Required Indicates where the section is placed on the page. The following choices are available in the list box:
----------------	---

- End Page
- First Page
- Last Page
- Next Page
- Same Page
- Whole Page

Print option	Status: Required Indicates whether the report is to be printed or not. The following choices are available in the list box:
--------------	--

- Not printed
- Printed]

Break sequence number	Status: Optional Describes the sequence of how break sections appear in the report. The maximum value is 9,999.
Break field	Status: Optional Contains the field in which a change in the data triggers the section printing. For example, to build a report listing customers and showing a subtotal for each state, sort the records by state and designate the state field as your break field. When your application calculates customer totals, the report prints a subtotal for each state. Type a field name up to a maximum of 30 characters, for break sections. Leave this property blank for sections other than break sections.
Break qualifier	Status: Optional Specifies the view path that includes the instance of the field named in the break field. Type a view name in the format <i>Root_View_Name.Sub_View_Name</i> .
Left margin	Status: Optional Specifies the left margin for that section. The maximum value is 9,999.

Set Contains Value Variant

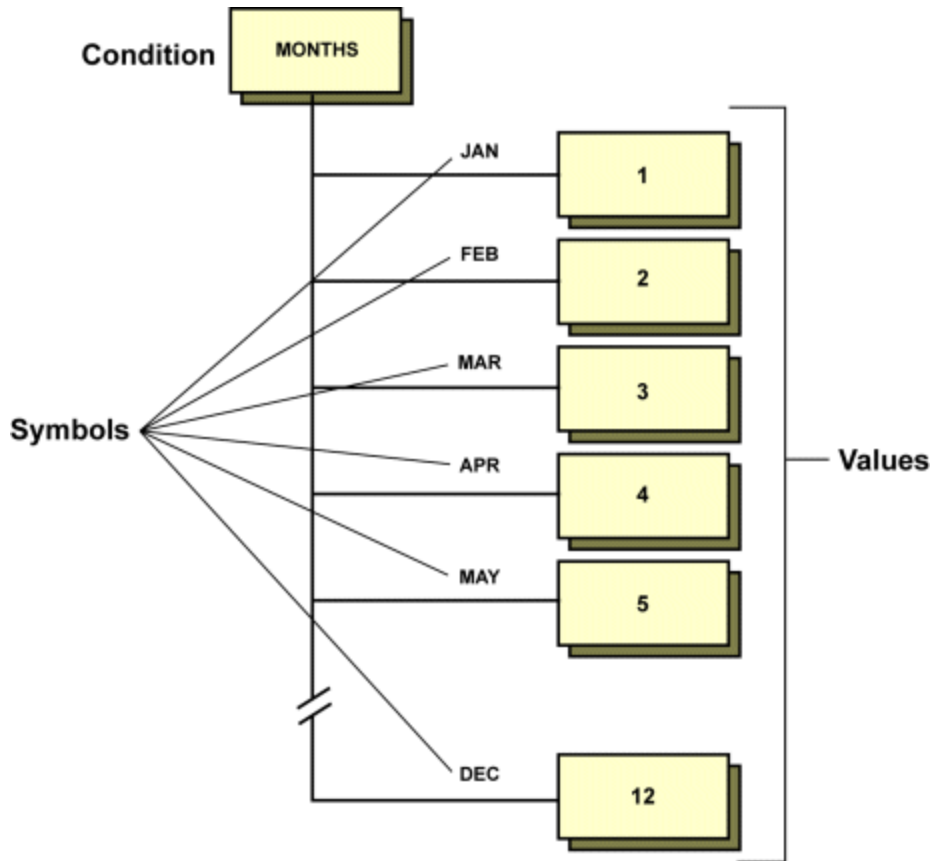
This variant ([Set contains Symbol Properties Dialog](#)) has a special importance in the AppBuilder environment. Its symbol property can be used as a substitute for the value entity type to which it relates, as shown in [Using Symbols](#). For example, many sets can contain the value 1. However, the relationship type between the value 1 and the set Months can have a symbol of "JAN," while the relationship type between the value 1 and the set Mode can have the symbol "on."

Set contains Symbol Properties Dialog

The screenshot shows a dialog box titled "Properties - [contains Symbol]". It has four tabs: "General", "Audit", "Remote Audit", and "Text". The "General" tab is active. Below the tabs are four input fields:

- Parent name: ACC_VERSION_TYPE
- Child name: AEXU5B_0
- Sequence number: 0
- Separator ID: 0

Using Symbols



Standards for the symbol and accompanying text for instances of the set contains a value relationship type depending on the usage of the set.

Set contains value Properties	
Property	Description
Symbol	Status: Required Can contain a maximum of 18 characters

Functional Sets

You can refer to the symbol of the values in a *functional set* to help make your Rules Language code more readable when you use system components. Sets are used as functional sets for local validation. That is, the symbols help describe or validate the meaning of the values where the values themselves are designed for a specific purpose. A set is a functional set when it contains values that standard system components use. Functional sets contain values that are codes the component needs to work correctly. In these cases, the symbol attached to the relationship describes the function of the value and can be used within rules code to refer to the value itself.

For example, the system component SET_FIELD_COLOR requires numeric data in its input view to work correctly. You can memorize the numbers associated with the different colors, or you can use a set that contains those numbers as values, and the meaning of those values as symbols. Rules Language accepts both methods.

map 2 to ATTR_COLOR of SET_FIELD_COLOR_I

or

map BLUE in COLORS to ATTR_COLOR of SET_FIELD_COLOR_I

Both of these statements map "2" to the field ATTR_COLOR. Using the symbol property in functional sets gives you a way to keep track of the meaning of numeric codes that system components use. Note that the symbol associated with a value does not explicitly depend on that value. A value equal to 1 could have many uses (and thus many associated symbols---but only one in a given set), because the relationship instance defines the use of the values, and the set itself defines the relationship instance. The text attached to the relationship instance should contain a more detailed description of the meaning of the symbol when necessary.

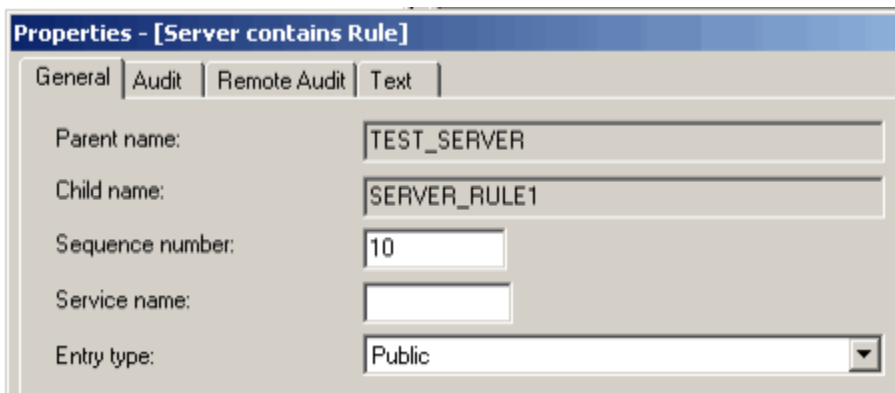
Setting Window Messages

Sets are used in association with setting window messages. The symbol property of the *contains* relationship type between the message set and a particular value provides a meaningful tag you can use to refer to the error set in the Rules Language code; the text associated with that *contains* relationship type provides the window message itself. AppBuilder provides a number of system components that developers can use to display pop-up message windows. You can use these components in conjunction with a message set to display pertinent information to your end user and thereby make your application interface friendlier. In a message set, each value represents the number of a particular message. You can display that message by using the appropriate system component, whether by number or by symbolic name. When you use sets in this way, make sure that the symbol and associated text conform to your standards for window messages. The symbol is used in the Rules Language code, and therefore should be a unique identifier describing the condition or action resulting in the message display. The text property of the *contains* relationship is the content of the window message. It is important that the text not only explain the condition that resulted in the window message, but also tell the user how to proceed. Additionally, consider the format of the text, because at runtime the window that contains the message is generated to accommodate the format as entered behind the Set contains a Value relationship type. Remember that the text that you type is exactly what the end user sees.

Server contains Rule Variant

This variant can occur only as Server contains Rule ([Server contains Rule Properties Dialog](#)).

Server contains Rule Properties Dialog



<i>Server contains Rule Properties</i>	
Property	Description
Service name	Status: Optional Maximum of 30 characters
Entry type	Status: Required The following choices are available in the list box:

- Private
- Public

When set to *Private*, the server root rule can only be called by a rule executing on the same machine. If a rule from another machine tries to call a Private server root rule, an AppBuilder runtime error is generated. The Private property can be used to isolate logical functionality within a machine without allowing access to that logical functionality from the outside of the target machine. Thus, a logical subsystem can be defined, allowing compartmentalized development and made accessible through the private interface, only to other rules residing on the same machine. When set to *Public*, the server root rule can be called from a rule executing on any machine.]

Converses (is-conversed-by)

Converses is a relationship type in which the first entity type causes the second entity type to be displayed on a screen, in an online application, or on paper in a batch application.

Example converses Properties Dialog

Properties - [Rule converses Window]

General | Audit | Remote Audit | Text

Parent name: HELLO_CLIENT

Child name: HELLO_WIN

Sequence number: 1

Separator ID: 0

Variants

- Rule converses Report
- Rule converses Window

Depends-on (is-depende-d-on-by)

Depends-on ([Depends-On Logical Process Properties Dialog](#)) is a relationship type in which interrelated processes and rules have an orderly sequence in which they are performed.

Depends-On Logical Process Properties Dialog

Properties - [depends-on Logical Process]

General | Audit | Remote Audit | Text

depends-on logical process

Parent name: CHR_G CHARGE_DEL

Child name: CHR_G STATUS_UPD

Sequence number: 0

Separator ID: 0

Description:

Depends on has the following properties:

- Status is Optional.
- Can contain a maximum of thirty characters.

Variants

- Logical Process depends-on Logical Process ([Depends-On Logical Process Properties Dialog](#))
- Process depends-on Process
- Rule depends-on Rule

Depends-on Logical Process Variant

The first variant can occur only as Logical Process depends-on Logical Process.

Process depends-on Process and Rule depends-on Rule Variants

These variants are identical except for their internal names.

- Process depends-on Process
- Rule depends-on Rule

Encapsulates (is-encapsulated-by)

Encapsulates ([Encapsulates Properties Dialog](#)) is a relationship in which the second entity type is included within the first entity type.

Encapsulates Properties Dialog

Properties - [Partition encapsulates]

General | Audit | Remote Audit | Text

Parent name:

Child name:

Sequence number: Separator ID:

Prepare time:

Object imp name:

Service name:

Significant time:

Server link type: Isolation mode:

Server owner:

Server qualifier:

Server plan name:

Collection ID:

Version ID:

Partition encapsulates Properties	
Property	Description
Prepare time	Reserved for future use
Object Imp name	Reserved for future use
Service name	Status: Optional Maximum of thirty characters
Significant time	Reserved for future use
Server link type	Status: Required The method of linking a rule that belongs to the associated server. The following choices are available in the list box: <ul style="list-style-type: none"> • <i>Dynamic</i> ---The code needed to resolve external references is shared with other rules and so any external references are resolved at runtime. • <i>Static</i> ---The code needed to resolve external references is copied into the executable load module of the rule at link time so any external references are resolved at prepare time.
Server owner	Status: Required The identifier according to which a DB2 plan or package is prepared for all rules that belong to the associated server. Maximum of thirty characters.
Server qualifier	Status: Required The identifier for all unqualified host SQL calls associated with the Partition unit. Maximum of thirty characters.
Server plan name	Status: Required The DB2 plan for the associated rules. Maximum of thirty characters.
Collection ID	Status: Optional The DB2 collection ID to be used for objects that are connected to this Partition unit

Version ID	Reserved for future use
Isolation Mode	<p>Status: Required</p> <p>The isolation mode to be used for objects that are connected to this Partition unit. The following choices are platform specific and are dependent on DB2 settings:</p> <ul style="list-style-type: none"> • Committed Read • Cursor Stability • Dirty Read • Repeatable Read

Variants

- Partition encapsulates Component
- Partition encapsulates Database
- Partition encapsulates File
- Partition encapsulates Function
- Partition encapsulates Machine
- Partition encapsulates Process
- Partition encapsulates Report
- Partition encapsulates Rule
- Partition encapsulates Set
- Partition encapsulates Server
- Partition encapsulates Window

Has (is-had-by)

Has ([Has Properties Dialog](#)) is a relationship type in which the first entity type owns the second entity type.

Has Properties Dialog

Table has Column Properties	
Property	Description
Null	Value can be Null or NotNull.
Distinct type	Statistic field to document the design, similar to the attribute of an Entity object. If this value is Absolute, then the value in Distinct is the number of rows that have unique values for this column. If this value is Ratio, then the value in Distinct is the percentage of rows that have unique values.
Distinct	Statistic field to document the design, similar to the attribute of an Entity object. If the Distinct type value is Absolute, then this value is the number of rows that have unique values for this column. If the Distinct type value is Ratio, then this value is the percentage of rows that have unique values.

Update percentage	The percentage of rows that update this column.
-------------------	---

Variants

- Application Configuration has Partition
- Business Object has State
- Entity has Identifier
- Entity has State
- Scheme has Value
- Function has Bitmap
- Key has Column
- Physical Event has Rule
- Process has Bitmap
- Report has Form
- Rule has Bitmap
- Table has Column
- Table has Key
- Window has Bitmap
- Window has Window Content

These variants are identical except for their internal names:

- Application Configuration has Partition
- Entity has Identifier
- Scheme has Value
- Key has Column
- Business Object has State
- Entity has State
- Table has Key

Has STD (has-a-state-transition-diagram)

Has STD is a relationship type in which the entity is associated with State Transition diagram.

Variants

- Business Object has-a-state-transition-diagram Drawing
- Entity has-a-state-transition-diagram Drawing
- State has-a-state-transition-diagram Drawing

Impacts (is-impacted-by)

Impacts is a relationship in which the first entity type affects the second.

<i>Process impacts Entity Properties</i>		
Property	Status	Description
Create flag	Optional	Specifies whether the instance of the first entity type creates the second.
Read flag	Optional	Specifies whether the instance of the first entity type reads the second.
Update flag	Optional	Specifies whether the instance of the first entity type updates the second.
Delete flag	Optional	Specifies whether the instance of the first entity type deletes the second.
Comments	Optional	Can contain a maximum of thirty characters.

Includes (is-included-in)

Includes ([Includes Properties Dialog](#)) is a relationship in which the second entity type is a part, an element, or a member of the first entity type.

Includes Properties Dialog

Properties - [View includes]

General | Audit | Remote Audit | Text

Parent name: HELLO_WIN_V

Child name: INPUT_MESSAGE

Sequence number: 10

Separator ID: 0

Null indicator: Nulls not Allowed

Occurs times: 000000

View includes Properties	
Property	Description
Null indicator	Status: Required The following choices are available in the list box: <ul style="list-style-type: none"> • Not Nulls With Default • Nulls Allowed • Nulls Not Allowed
Occurs times	Status: Required In the View includes Field relationship type, this property can be set only to 0 or 1; in View includes View relationship type, this property can contain a maximum value of 999,999.

Variants

- View includes Field
- View includes View

Influences (is-influenced-by)

Influences is a relationship in which the first entity type affects the second. The Influences (is-influenced-by) relationship has the following properties:

- Status is Optional.
- Can contain a maximum of thirty characters.

Event Influences Business Object Properties		
Property	Status	Description
Description	Optional	Can contain a maximum of 30 characters.

Variants

Event influences Business Object

Initiates (is-initiated-by)

Initiates is a relationship in which the first entity type invokes the second. The Initiates (is-initiated-by) relationship has the following properties:

- Status is Optional.
- Can contain a maximum of thirty characters.

Logical Process initiates Event Properties		
Property	Status	Description
Descriptions	Optional	Can contain a maximum of 30 characters.

Variants

Logical Process initiates Event

Intersects (intersects-with)

Intersects is a relationship in which the first entity type interacts in some fashion with the second entity type.

Variants

Function intersects-with Entity

Function intersects-with Entity Properties	
Property	Description
Create flag	Status: Optional Specifies whether the instance of the first entity type creates the second
Read flag	Status: Optional Specifies whether the instance of the first entity type reads the second
Update flag	Status: Optional Specifies whether the instance of the first entity type updates the second
Delete flag	Status: Optional Specifies whether the instance of the first entity type deletes the second
Comments	Status: Optional Can contain a maximum of 30 characters

Invokes (is-invoked-by)

Invokes is a relationship in which the first entity type invokes the second.

Variants

- Window invokes Rule
- Window invokes Component

Is-carried-out-at (Carries Out)

Is-carried-out-at refers to the place where a function or process is performed.

Is-carried-out-at location Properties	
Property	Description
Major involvement in function	Status: Optional Specifies how the instance is carried out at the location
Minor involvement in function	Status: Optional Specifies how the instance is carried out at the location
Comments	Status: Optional Can contain a maximum of thirty characters

Is-composed-of (Composes)

Is-composed-of ([Example is-composed-of Properties Dialog](#)) is a relationship in which the first entity is made up of constituent parts, one of which is represented by the second entity.

Example is-composed-of Properties Dialog

Properties - [Identifier is-composed-of Attribute]			
General	Audit	Remote Audit	Text
Parent name:	<input type="text" value="GD_DEPARTMENT"/>		
Child name:	<input type="text" value="GD_DEPT_CODE"/>		
Sequence number:	<input type="text" value="10"/>		
Separator ID:	<input type="text" value="0"/>		

<i>Attribute is-composed-of Attribute and Data Type is-composed-of Data Type</i>	
Property	Description
Occurs	Status: Optional Maximum value of 999,999

Variants

- Attribute is-composed-of Attribute
- Attribute is-composed-of Relationship
- Data Type is-composed-of Data Type
- Event is-composed-of Event
- Logical Process is-composed-of Logical Process
- Identifier is-composed-of Attribute
- Identifier is-composed-of Relationship

Is-constrained-by (Constrains)

Is-constrained-by ([Data Type Is-Constrained-By Set Properties Dialog](#)) the data described in a data type is limited by the data that a set contains.

Data Type Is-Constrained-By Set Properties Dialog

Properties - [Data Type is-constrained-by Set]			
General	Audit	Remote Audit	Text
Parent name:	<input type="text" value="DATATYPE1"/>		
Child name:	<input type="text" value="TEST_DATATYPE_SET"/>		
Sequence number:	<input type="text" value="10"/>		
Separator ID:	<input type="text" value="0"/>		

Variants

Data Type is-constrained-by Set.

Is-defined-by (Defines)

Is-defined-by ([Example is-defined-by Properties Dialog](#)) is a relationship type in which a leaf process has a single rule that executes the logic the leaf process represents.

Example is-defined-by Properties Dialog

Properties - [Process is-defined-by]

General | Audit | Remote Audit | Text

Parent name: AGENT MAINTENANCE

Child name: AGENT_DET_DIS

Sequence number: 10

Separator ID: 0

Variants

Process is-defined-by Rule

Is-described-by (Describes)

Is-described-by ([Example is-described-by Properties Dialog](#)) is a relationship type in which the second entity type helps to define the first entity type.

Example is-described-by Properties Dialog

Properties - [is-described-by Attribute]

General | Audit | Remote Audit | Text

Parent name: GD_EMPLOYEE

Child name: GD_EMP_NAME

Sequence number: 10

Separator ID: 0

Optional

The Entity is-described-by Attribute relationship has the following properties:

- Status is Optional.
- Whether the second entity type is Status: Required by the first entity type is optional.

Variants

- Entity is-described-by Attribute
- Relationship is-described-by Attribute

Is-distributed-at (Where-is-distributed)

Is-distributed-at describes the distribution of data at a given location.

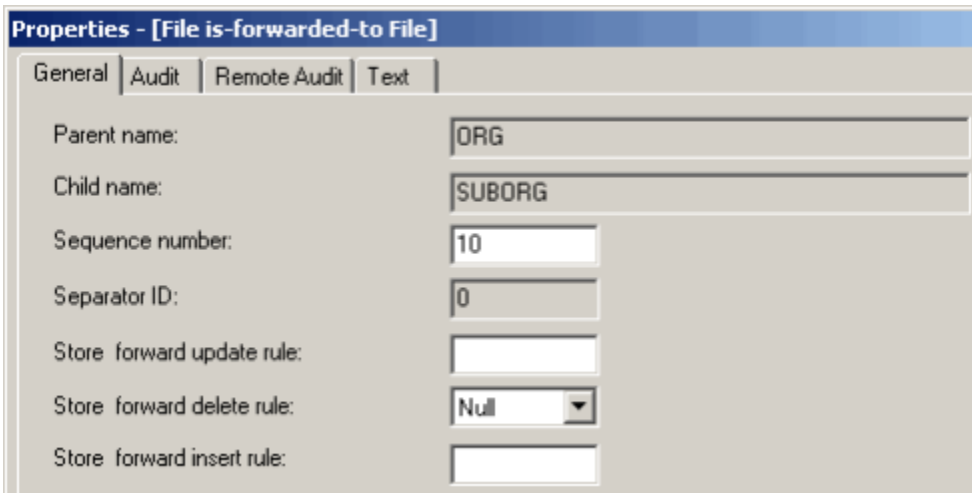
Entity is-distributed-at Location Properties	
Property	Description
Master indicator	Status: Optional Indicates whether the data are master
Variant indicator	Status: Optional Indicates whether the data are variant
Partitioned indicator	Status: Optional Indicates whether the data are partitioned

Replicated indicator	Status: Optional Indicates whether the data are replicated
Subset data	Status: Optional Indicates whether the data are subset
Reorganized data	Status: Optional Indicates whether the data are reorganized
Teleprocessing (remote data)	Status: Optional Indicates whether the data are remote
Comments	Status: Optional Can contain a maximum of thirty characters

Is-forwarded-to (Is-forwarded-from)

Is-forwarded-to ([File Is-forwarded-to File Properties Dialog](#)) refers to the interrelationship between files on different hardware platforms.

File Is-forwarded-to File Properties Dialog



File is-forwarded-to File Properties	
Property	Description
Store & forward update rule	Status: Optional Used for the Store and Forward System to send data from the source file to the target file. The default for these properties is 0. Other characters available are the suffixes of the name of the ISPF skeleton of the piece of code to be called in. Each property can contain a maximum of 1 character.
Store & forward delete rule	Status: Optional See Store & Forward Update Rule property
Store & forward insert rule	Status: Optional See Store & Forward Update Rule property

Variants

File is-forwarded-to File

Is-implemented-by (Implements)

Is-implemented-by ([Is-implemented-by Properties Dialog](#)) is a relationship in which the first (logical) entity is converted through the forward engineering process into the second (physical) entity.

In the enterprise repository, you might see this relationship type listed under more entity types than shown here. Disregard any others and do not use them. They are reserved for future use and are not supported.

While you can view this relationship type, do not modify it. Doing so can damage the results of the forward engineering process.

Is-implemented-by Properties Dialog

Properties - [Logical is-implemented-by Relationship]

General | Audit | Remote Audit | Text

is-implemented-by

Parent name:

Child name:

Sequence number:

Separator ID:

Variants

- Attribute is-implemented-by Collection
- Attribute is-implemented-by Column
- Attribute is-implemented-by File
- Attribute is-implemented-by Key
- Attribute is-implemented-by Table
- Collection is-implemented-by Collection
- Collection is-implemented-by Column
- Collection is-implemented-by File
- Collection is-implemented by Key
- Collection is-implemented-by Table
- Data type is-implemented-by Collection
- Data type is-implemented-by Column
- Data Type is-implemented-by File
- Data Type is-implemented-by Key
- Data Type is-implemented-by Table
- Entity is-implemented-by Collection
- Entity is-implemented-by Column
- Entity is-implemented-by File
- Entity is-implemented-by Key
- Entity is-implemented-by Table
- Identifier is-implemented-by Collection
- Identifier is-implemented-by Column
- Identifier is-implemented-by File
- Identifier is-implemented-by Key
- Identifier is-implemented-by Table
- Logical Process is-implemented-by Rule
- Relationship is-implemented-by Collection
- Relationship is-implemented-by Column
- Relationship is-implemented-by File
- Relationship is-implemented-by Key
- Relationship is-implemented-by Table
- Table is-implemented-by Collection
- Table is-implemented-by Column
- Table is-implemented-by File
- Table is-implemented-by Key
- Table is-implemented-by Table

Is-involved-in (Involves)

Is-involved-in ([Organization Is-Involved-In Properties Dialog](#)) is a relationship type in which the first entity is connected by association or participation to the second entity.

Organization Is-Involved-In Properties Dialog

The dialog box shows the following values:

- Parent name: ORG1
- Child name: TEST_PROJECT
- Sequence number: 0
- Separator ID: 0
- Responsible: No
- Major_Inv: Yes
- Minor_Inv: No
- Comments: (empty)

Is-involved-in Relationship Type Properties	
Property	Description
Responsible	Status: Required Whether the entity is responsible for causing the logical process
Major_INV	Status: Required Whether the entity that has a major involvement with the logical process
Minor_INV	Status: Required Whether the entity that has a minor involvement with the logical process
Comments	Status: Optional Can contain a maximum of thirty characters

Is-keyed-by (Keys)

Is-keyed-by is a relationship in which a field points to or serves as a key to access a file.

Is-keyed-by field Properties	
Property	Description
Update key	Status: Required The following choices are available in the list box: <ul style="list-style-type: none"> • Delete Key • Update Key

Variants

File is-keyed-by Field

Is-modified-at (Where-is-modified)

Is-modified-at describes the action carried out on an entity at a given location.

Entity is-modified-at Location Properties	
Property	Description
Create flag	Status: Optional Specifies whether the instance of the first entity type creates the second
Read flag	Status: Optional Specifies whether the instance of the first entity type reads the second
Update flag	Status: Optional Specifies whether the instance of the first entity type updates the second
Delete flag	Status: Optional Specifies whether the instance of the first entity type deletes the second
Comments	Status: Optional Can contain a maximum of 30 characters

Is-modified-by (Modifies)

Is-modified-by ([Example is-modified-by Properties Dialog](#)) is the relationship type in which the first entity is changed in form or character by the second entity.

Example is-modified-by Properties Dialog

The screenshot shows a software dialog box titled "Properties - [Entity Modified-By]". It features four tabs: "General", "Audit", "Remote Audit", and "Text". The "General" tab is selected. The dialog contains several input fields and dropdown menus:

- Parent name:** A text box containing "ENTITY_1".
- Child name:** A text box containing "LOCATION_1".
- Sequence number:** A text box containing "0".
- Separator ID:** A text box containing "0".
- Create:** A dropdown menu with "Yes" selected.
- Read:** A dropdown menu with "No" selected.
- Update:** A dropdown menu with "No" selected.
- Delete:** A dropdown menu with "Yes" selected.
- Comments:** An empty text box.

Is-modified-by Properties	
Property	Description
Create	Status: Required The following choices are available: <ul style="list-style-type: none"> • Yes • No Whether the entity is altered in this manner

Read	Status: Required The following choices are available: <ul style="list-style-type: none"> • Yes • No Whether the entity is altered in this manner
Update	Status: Required The following choices are available: <ul style="list-style-type: none"> • Yes • No Whether the entity is altered in this manner
Delete	Status: Required The following choices are available: <ul style="list-style-type: none"> • Yes • No Whether the entity is altered in this manner
Comments	Status: Optional Can contain a maximum of thirty characters

Variants

Entity is-modified-by Logical Process

Is-preconditioned-by (Preconditions)

Is-preconditioned-by ([Example is-preconditioned-by Properties Dialog](#)) is the relationship type in which the second entity type is a prerequisite for the first entity type.

Example is-preconditioned-by Properties Dialog

Transition is-preconditioned-by State Properties	
Property	Description
Description	Status: Optional Can contain a maximum of thirty characters

Variants

Transition is-preconditioned-by State

Is-profiled-by (Profiles)

Is-profiled-by is a relationship type in which the group type and migration type relate.

Variants

Migration is-profiled-by Group

Is-referenced-by (References)

Is-referenced-by ([Example is-referenced-by Properties Dialog](#)) is a relationship type in which the first entity uses the second entity for determining the source of columns within a database.

Example is-referenced-by Properties Dialog

Properties - [Table is-referenced-by Key]

General | Audit | Remote Audit | Text

table is-referenced-by key

Parent name: ABTMT_STG_TYP

Child name: ABS_TO_PAB_FK

Sequence number: 10

Separator ID: 0

Referential integrity

<i>Table is-referenced-by Key Properties</i>	
Property	Description
Referential integrity	Status: Required Whether referential integrity is enforced

Variants

Table is-referenced-by Key

Is-related-via (Relates-to)

Is-related-via ([Is-related-via Properties Dialog](#)) relationship type specifies the nature of a relationship.

Is-related-via Properties Dialog

Properties - [is-related-via Relationship]

General | Audit | Remote Audit | Text

Parent name:

Child name:

Sequence number:

Separator ID:

Cardinality: Option Dependent

Role:

Minimum cardinality:

Maximum cardinality:

<i>Is-related-via Properties</i>	
Property	Description
Cardinality	Status: Required The following choices are available in the list box: <ul style="list-style-type: none"> • 1 • Many • Unknown This property is not applicable if the relationship entity type's Complex Indicator property is SUBTYPE.
Chief executive	Status: Optional Indicates whether the relationship is controlling
Dependent	Status: Optional Indicates whether the relationship is dependent
Option	Status: Optional Indicates whether the relationship is optional
Abstract	Status: Optional Indicates whether the relationship is abstract. Applicable only if the relationship entity type's Complex Indicator property is SUBTYPE.
Role	Status: Optional Can contain a maximum of 30 characters
Minimum cardinality	Status: Optional Can contain a numeric value from 0 to 99,999
Maximum cardinality	Can contain a numeric value from 0 to 9,999,999,999

Variants

Entity is-related-via Relationship

Is-responsible-for (is-the-responsibility-of)

Is-responsible-for is the relationship type in which the first entity is held accountable for the second entity.

Is-site-of (Has-as-its-site)

Is-site-of is a relationship type in which the location is the physical location where the second entity takes place.

Is-typed-by (Types)

Is-typed-by ([Attribute Is-typed-by Data Type Properties Dialog](#)) allows you to give an property a data type.

Attribute Is-typed-by Data Type Properties Dialog

The screenshot shows a dialog box titled "Properties - [Attribute is typed-by Data Type]". It has four tabs: "General", "Audit", "Remote Audit", and "Text". The "General" tab is selected. The dialog contains the following fields:

- Parent name: ACTUAL_RET_CNT
- Child name: ACTUAL_RET_CNT
- Sequence number: 10
- Separator ID: 0

Variants

Attribute is-typed-by Data Type

Maps to (is-mapped-to)

Maps to is a relationship type in which a conceptual business model object has a mapping relationship to a logical business model object.

<i>Maps to Relationships</i>		
From		To
Value	Maps To	Entity
		Logical Process
		Attribute
		Event
		Relationship

<i>Maps to Properties</i>	
Property	Description
MapsTo_Type	Status: Optional Can contain a maximum of 1 character Default is 3 (Undefined Format) mDomMapsToType <ul style="list-style-type: none"> • Not Defined • Indirect • Direct • Customization

Member (is-member-of)

Member is a relationship type that defines membership in a grouping.

Member Properties	
Property	Description
MemberType	Status: Optional Can contain a maximum of 1 character Default is 3 (Undefined Format) mDomMemberType <ul style="list-style-type: none"> • Member • Focal • Supertype • Neighbor

Variants

Business Object has member Entity

Owns (is-owned-by)

Owns ([Owns View Properties Dialog](#)) is a relationship in which the second entity type is attached to the first.

Variants

This relationship type has two variants:

- [Business Object Owns Variant](#)
- [Owns View Variant](#)

Business Object Owns Variant

- Business Object owns Entity
- Business Object owns Event
- Business Object owns Logical Process

Business Object owns Properties Dialog

The screenshot shows a dialog box titled "Properties - [Business Object owns]". It has four tabs: "General", "Audit", "Remote Audit", and "Text". The "General" tab is active. There are four input fields: "Parent name" with the value "VEHICLE", "Child name" with the value "LOCATION", "Sequence number" with the value "10", and "Separator ID" with the value "0".

Owns View Variant

The view that is owned-by Entity type defines its data structure (or part of its data structure). For example, a rule might have an input view, an output view, or both. This variant can occur as:

- Component owns View
- File owns View
- Physical Event owns View
- Report owns View
- Rule owns View
- Section owns View
- Window owns View

Owns View Properties Dialog

Properties - [owns View]

General | Audit | Remote Audit | Text

Parent name:

Child name:

Sequence number:

Separator ID:

View usage:

<i>Owns view Properties</i>	
Property	Description
View usage	<p>Status: Required The following choices are available in the list box to specify how a rule, file, or component uses the view:</p> <ul style="list-style-type: none"> • Alternative View • Candidate Index View • Data View • Foreign Index View • Global View • Input & Output View • Input View • Output View • Primary Index View • Work View

Partitioned (partitioned-by)

A partition can have multiple Application Configurations. The Application Configuration contains partitions allowing the Function to be prepared (partitioned) to different machines.

Partitioned-by Properties Dialog

Properties - [Function partitioned-by Configuration]

General | Audit | Remote Audit | Text

Parent name:

Child name:

Sequence number:

Separator ID:

Variants

Function partitioned-by Configuration

Redefines

One view can redefine another view. The data contained in the two views are stored at the same address in memory. Essentially, the two views are just different names for the same collection of data allowing you to use multiple definitions for the same memory space. This is an alternative to overlaying views, which copies the data from one area in memory to another thus creating two copies of the same data.

Follow these steps to have a view redefine another view:

Open the Construction Workbench.

1. From the Hierarchy window in Construction Workbench display a hierarchy with two Views.
2. Create a "View Includes View" relationship between the two views, making the original view as the parent and the redefined view as the child.
3. Right-click the child view and select *Relationship Properties*.
4. Change the Null indicator property to Redefines View.

The following restrictions apply to using redefined views:

- The first view cannot be a locally-declared view.
- The length of the second view must be less than or equal to the length of the first view. You will encounter errors if the second view exceeds the length of the first view.

Variants

View Redefines View

Refers-to (is-referred-to-by)

The Refers-to relationship type describes the data associated with another entity. For example, when a rule refers-to a set, the values associated with the set are available to that rule.

Variants

- Component refers-to Set
- Field refers-to Set
- Report refers-to Set
- Rule refers-to Set
- Window refers-to Set

Refines-into (is-refined-from)

Refines-into ([Process Refines-Into Process Properties Dialog](#)) is a relationship type in which the first entity can be broken down or decomposed into the second entity.

Process Refines-Into Process Properties Dialog

<i>Process refines-into Process Properties</i>	
Property	Description
Condition flag	Status: Optional Can contain up to four alphanumeric characters

Variants

- Business Object refines-into Business Object
- Function refines-into Process
- Process refines-into Process

The third variant can occur only as Process refines-into Process ([Process Refines-Into Process Properties Dialog](#)).

Replaces System (System is-replaced-by)

Replaces is the relationship type in which the first entity type (typically part of a newly-developed system) supplants all or part of the second entity type.

<i>Process Replaces System Properties</i>	
Property	Description
Currently supported	Status: Optional Indicates whether the instance is currently supported
Planned support	Status: Optional Indicates whether support for the instance is planned
Comments	Status: Optional Can contain a maximum of thirty characters

Results-in (is-the-result-of)

Results-in is a relationship type in which the second entity type is a product of the first entity type.

<i>Transition Results-in State Properties</i>	
Property	Description
Description	Status: Optional Can contain a maximum of 30 characters

Variants

Transition results-in State

Stores (provides-storage-for)

Stores is the relationship type used to define that one entity provide storage for another entity.

<i>Stores Properties</i>	
Property	Description
BusDataFormat	Status: Optional Can contain a maximum of 1 character Default is 3 (Undefined Format) mDomBusDataFormat <ul style="list-style-type: none"> • Character • Numeric • Code • Undefined Format • Date • Time
BusDataLen	Status: Optional Can contain a maximum of five characters Default is 0015
BusDataFrac	Status: Optional Can contain a maximum of two characters
ClassInstnc	Status: Optional Can contain a maximum of thirty characters

FormatOR	Status: Optional Can contain a maximum of one character Default is N mDomBoolean <ul style="list-style-type: none"> • No • Yes
ImpAbbr	Status: Optional Can contain a maximum of twenty characters
ImplFormatType	Status: Optional Can contain a maximum of one character Default is C mDomImplType <ul style="list-style-type: none"> • Date • VarChar • Graphic Character (DBCS) • Character • Picture • Mixed Character • Small Integer or Integer • Decimal • Timestamp • Time • Image • Undefined • Code • Class Instance Ref • Class Instance Value
Fld_Len	Status: Optional Can contain a maximum of five characters Default is 15
Fld_Frac	Status: Optional Can contain a maximum of two characters Default is 0

Supports (Is-supported-by)

Supports ([Organization Supports Process Properties Dialog](#)) is a relationship type in which the first entity type executes the activities of the second entity type.

Organization Supports Process Properties Dialog

Properties - [Organization supports Process]

General | Audit | Remote Audit | Text

Parent name:

Child name:

Sequence number:

Separator ID:

Responsible

Major involvement in function

Minor involvement in function

Comments:

Organization supports Function and Organization supports Process Properties	
Property	Description
Responsible	Status: Optional Can contain a maximum of one character
Major involvement in function	Status: Optional Specifies how the instance is supported at the location
Minor involvement in function	Status: Optional Specifies how the instance is supported at the location
Comments	Status: Optional Can contain a maximum of thirty characters

Organization Supports Entity Properties Dialog

The screenshot shows a dialog box titled "Properties - [Organization supports Entity]". It has four tabs: "General", "Audit", "Remote Audit", and "Text". The "General" tab is active. The fields are as follows:

- Parent name: KS_ORG
- Child name: TESTENT
- Sequence number: 0
- Separator ID: 0
- Four checkboxes: Create flag, Read flag, Update flag, and Delete flag (all are unchecked).
- Comments: (empty text box)

Organization supports Entity Properties	
Property	Description
Create flag	Status: Optional Specifies whether the instance of the first entity type creates the second
Read flag	Status: Optional Specifies whether the instance of the first entity type reads the second
Update flag	Status: Optional Specifies whether the instance of the first entity type updates the second
Delete flag	Status: Optional Specifies whether the instance of the first entity type deletes the second
Comments	Status: Optional Can contain a maximum of thirty characters

Triggers (is-triggered-by)

Triggers is the relationship type in which the first entity type causes the second entity type to be invoked.

Event Triggers Process Properties	
Property	Description

Inclusive flag	Status: Required The following choices are available in the list box: <ul style="list-style-type: none"> • Exclusive • Inclusive • Regular
Exclusive/ or sequence number	Status: Required Can contain a numeric value from 0 to 999
Description	Status: Optional Can contain a maximum of thirty characters

Triggers Logical Process Properties Dialog

Properties - [triggers Logical Process]

General | Audit | Remote Audit | Text

Parent name: TEST_EVENT1

Child name: TEST_PROCI

Sequence number: 0

Separator ID: 0

Description:

Variants

- Event triggers Process
- Event triggers Logical Process
- Transition triggers Logical Process
- Rule triggers Physical Event

The first variant can occur only as Event triggers Process.
The second variant can occur as:

- Event triggers Logical Process
- Transition triggers Logical Process

Uses (is-used-by)

Uses ([Rule Uses Rule Properties Dialog](#)) is a relationship type in which the first entity type invokes the second.

Rule Uses Rule Properties Dialog

Properties - [Rule uses Rule]

General | Audit | Remote Audit | Text

Parent name: HELLO_MIDTIER

Child name: QA_COOP_DATA_TYPE_CHAR_40

Sequence number: 60

Separator ID: 0

Variants

- Rule uses Component
- Rule uses Rule

Text Property Conventions

Text Property Conventions

Each object has a text property. This is a free-form text file that serves two purposes:

- Object specification
- On-screen end-user help for windows and fields

You must create and maintain the text files for objects that you create or change. You can associate descriptive text with each object. By maintaining documentation as properties of the objects themselves, you ensure that it is as current as the objects are. The objects and their descriptions are maintained in synchronization. Guidelines include:

- [General Guidelines](#)
- [Abbreviation Guidelines](#)

Objects that you create or modify can be reused in other applications. To promote efficiency and avoid redundancy, accurately document the true purpose or function of the object. If your text is not sufficiently specific, other developers might duplicate your work inadvertently, instead of properly reusing your object.

The following sections discuss the information you should include in text files.

General Guidelines

The text you write provides information about the object and its modifications. The text must provide enough information for other developers to determine whether they can reuse the object in a new application. Include at least the following information about each object instance.

Object Name

What the object is called.

Narrative

A description, written in business language, of what the object is or does and how the system uses it.

Special Considerations

Information about the object that a developer who is considering reusing the object might need to know, including any limitations that might restrict its reuse in other applications. Also include information about the object's future use or updates.

History

The maintenance history of the object, which includes any changes made to the object since it was created and why these changes were made.

Object-specific Characteristics

Any special characteristics that apply to some objects. Include a short explanation of each. For example, an entity can have subtypes listed here.

Abbreviation Guidelines

Because of limitations regarding the length of names, there are times when abbreviations are required. Use the following guidelines when you create abbreviations.

- Limit abbreviations to a maximum of six characters.
- Do not use special characters in abbreviations.
- Do not abbreviate words containing five or fewer characters.
- Eliminate vowels, if possible, except for the first letter of the word.
- If a word contains double consonants, eliminate one of each pair, starting at the end of the word.
- If the word is still longer than six characters, eliminate characters that have no graphic or phonetic value in conveying meaning, starting at the end of the word.
- Abbreviate all descriptive words when at least one descriptive word must be abbreviated.

Enterprise Administration Information Model

Enterprise Administration Information Model

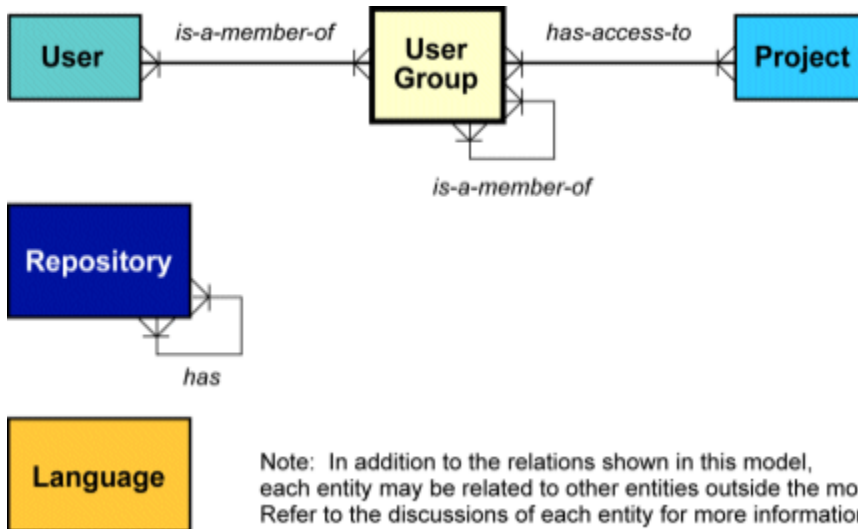
The Enterprise Administration Information Model is a subset of the overall AppBuilder Information Model. It includes entities and relationships that define the administration of the enterprise repository.

Administration Information Model

An Information Model is a collection of entity and relationship types, collectively called object types. Use the object types in the Administration Information Model to determine user access and to define the repository.

In Information Model diagrams, boxes represent entities, and lines connecting the boxes represent relationships. Because a relationship always connects two entities, you can view the relationship from the perspective of either entity. For example, user *is-a-member-of* group or group *has-as-a-member* user are two different ways of looking at the same relationship. While both perspectives indicate that *user* is a subordinate (or child) entity of *group*, the use of the parent name to child name relationship perspective is preferred. The Administration Information Model, shown in [The Administration Information Model](#), defines *many-to-many* relationships between several entities.

The Administration Information Model



The Administration Information Model includes the user, group, project, language, and repository entity types. Use the user, group, and project entities to define repository access, and the repository and language entities to define the repository itself.

You can view the Administration Information Model entities through the Administration (ADM) project.

In addition to seeing the Administration Information Model entities listed in the ADM project, you will see a migration entity. See the [Enterprise Migration Guide](#) for more information about migrations.

Select the ADM project to *create*, *read*, *update*, and *delete* all instances of the administrative entities and their relationships. Do not perform the other repository maintenance tasks within the AppBuilder environment.

User Entity

The user entity defines a user to the AppBuilder environment. Prior to being defined, the user must have a valid TSO user ID.

Whenever you add an entity, you must assign values for several attributes. Each user entity has two types of attributes: user and audit. [User entity attributes](#) describes each user entity attribute.

Unless the attribute descriptions specify otherwise, blanks are not allowed.

User entity attributes		
Attribute	Requirements	Description
Name	Required	A unique alphanumeric string up to thirty characters that identifies the user. Usually the user's first initial and last name. The names should be consistent to facilitate searching and administration.

User Name	Optional	An alphanumeric string up to thirty characters, including blanks. Usually the user's full name.
User ID	Required	A unique alphanumeric string up to seven characters, which must match the user's TSO user ID.
Job Accounting Information	Required	An alphanumeric string up to thirty characters, including blanks, that specifies accounting information such as billing for CPU time.
User Location	Optional	An alphanumeric string up to eight characters that specifies where the user is located in the organization, such as department or city.

The system generates all of the audit attributes, which are the same for all entities. The audit attributes consist of fourteen fields that indicate who created or modified the object, as well as the time and date it was created or last modified. You cannot directly edit these field values. [Entity Audit Attributes](#) describes the entity audit attributes.

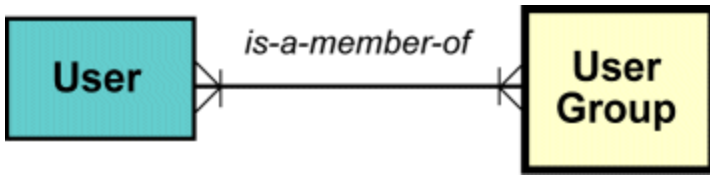
Entity Audit Attributes	
Attribute	Description
Date Created	The date (year/month/day) the entity was created.
Time Created	The time, based on a 24-hour clock, the entity was created.
Created by	The user ID of the individual who created the entity.
Date Maintained	The date (year/month/day) the entity was last modified.
Time Maintained	The time, based on a 24-hour clock, the entity was last modified.
Maintained by	The user ID of the individual who last modified the entity.
Project	The name of the project in which the entity was created.
Change #	The number of times that the entity has been updated. A value of 0001 indicates the entity has not changed since it was created and 0002 indicates it has been changed and saved once since it was created. <i>NOTE:</i> The change number increases by one each time an object is saved, even if the object data does not change, or if only its parts change.
Owner ID	The user ID of the person who is responsible for the entity.
Local Maintenance Date	The date (year/month/day) the entity was last modified.
Local Maintenance Time	The time, based on a 24-hour clock, the entity was last modified.
Local Maintenance Project	The name of the project in which the entity was created.
Local Maintenance User	The user ID of the person who last modified the entity.
Version	The repository version in which the entity was created.

An instance of a user entity can have the following relationships:

- *Is-a-member-of* Group
- *Is-a-root-of* Migration

A user must be related to a group to have access to the repository (see [User is-a-member-of Group](#)).

User is-a-member-of Group



When you define a user to a group, that user has access to the same objects in projects the group can access. You determine the level of access when you define the relationship attribute. [Is-a-member-of Group Relationship Attribute](#) describes the attribute for the *is-a-member-of* Group relationship.

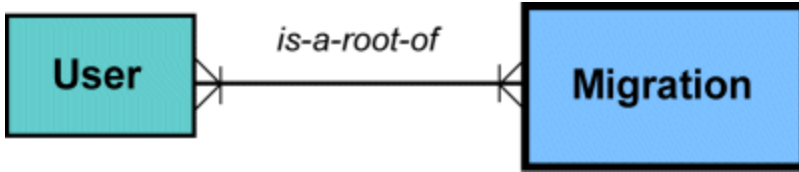
Is-a-member-of Group Relationship Attribute		
Attribute	Status	Description
User Authority Level	Required	One of the following: <ul style="list-style-type: none"> • Project Leader • Analyst • Programmer • Auditor

Just as the system automatically generates audit attributes for the entity, the system also generates similar audit attributes for relationships. [Relationship Audit Attributes](#) describes the relationship audit attributes, which are common to all relationships.

Relationship Audit Attributes	
Attribute	Description
Date Created	The date (year/month/day) the relationship was created.
Time Created	The time, based on a 24-hour clock, the relationship was created.
Created by	The user ID of the person who created the relationship.
Date Maintained	The date (year/month/day) the relationship was last modified.
Time Maintained	The time, based on a 24-hour clock, the relationship was last modified.
Maintained by	The user ID of the person who last modified the relationship.
Change #	The number of times the relationship has been updated. A value of 0001 indicates the relationship has not changed since created and 0002 indicates it was changed and saved once since created. <i>NOTE:</i> The change number increases by one each time an object is saved, even if the object data does not change, or if only its parts change.
Local Maintenance Date	The date (year/month/day) the relationship was last modified.
Local Maintenance Time	The time, based on a 24-hour clock, the relationship was last modified.
Local Maintenance Project	The name of the project in which the relationship was created.
Local Maintenance User	The user ID of the individual who last modified the relationship.
Version	The project version in which the relationship was created.

Just as you can migrate objects between repositories and versions, you can migrate users (see [User is-a-root-of Migration](#)).

User is-a-root-of Migration



When you create a user in one repository, you can migrate the user entity to other repositories rather than create a new user entity in each repository. When you create a relationship between a user and a migration entity, that user becomes part of a migration package. You specify that only the entity is to be migrated when you define the relationship attribute. (See [Group Entity](#) for information on migrating groups of users.) [Is-a-root-of Migration Relationship Attribute](#) describes the attribute for the *is-a-root-of* Migration relationship.

Is-a-root-of Migration Relationship Attribute		
Attribute	Status	Description
Scope Type	Required	"Entity Only" from the following: <ul style="list-style-type: none"> • Entity only • One level • Prepare hierarchy • Entity hierarchy • Drawing

Group Entity

Use the Group entity to create teams of users who are allowed access to a specific set or sets of objects in the repository. Thus, a group entity has a subset *users*.

Each group entity has two types of attributes: group and audit. [Group entity attributes](#) describes each group entity attribute.

Unless specified otherwise, blanks are not allowed.

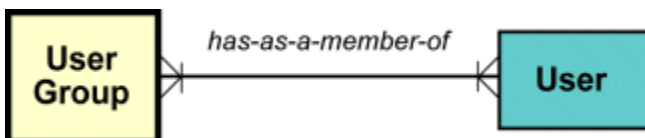
Group entity attributes		
Attribute	Status	Description
Name	Required	A unique alphanumeric string up to thirty characters that identifies the group.
Description	Optional	An alphanumeric string up to thirty characters, including blanks, that describes the purpose of the group.

An instance of a group entity can have the following relationships:

- *Has-as-a-member* User
- *Has* Project
- *Is-a-member-of* Group
- *Has-as-a-member* Group
- *Is-a-root-of* Migration
- *Is-profiled-by* Migration
- *Is-profiled-by* Rebuild Package

Users can access objects in projects only through groups (see [Group has-as-a-member User](#)).

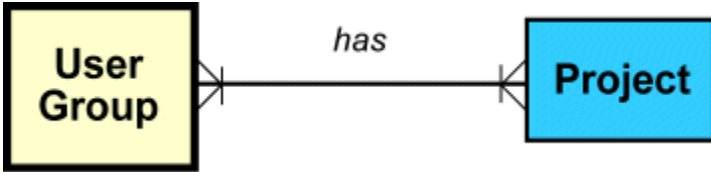
Group has-as-a-member User



Group *has-as-a-member* User is just another perspective of the User *is-a-member-of* Group relationship. See [Is-a-member-of Group Relationship Attribute](#) for a description of the relationship attributes.

The group relationship to a project—a set of objects—gives the members of the group access to objects in the project. See [Group has Project](#).

Group has Project

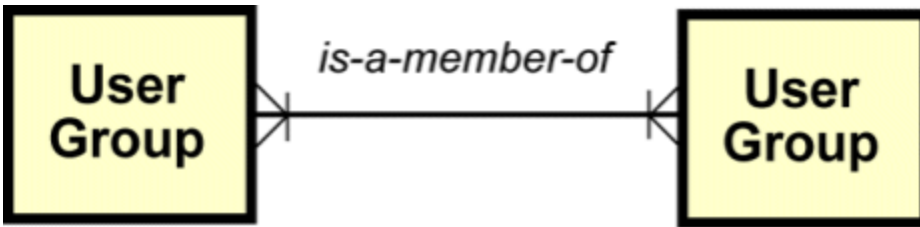


With the group to project relationship attribute, you can apply security features that limit the actions allowed by the relationship. [Has project Relationship Attributes](#) describes each attribute for the *has-project* relationship.

Has project Relationship Attributes		
Attribute	Status	Description
Scope Mode	Required	One of the following choices to indicate whether the Method Name and Scope attributes are inclusive or exclusive: <ul style="list-style-type: none"> • Include • Exclude

To further define group--and therefore user--access, you can make groups members of groups by relating them to each other (see [Group is-a-member-of Group](#)).

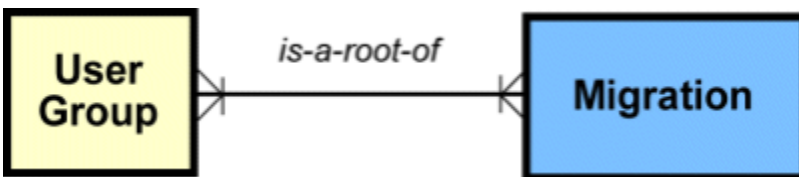
Group is-a-member-of Group



The *is-a-member-of* relationship indicates that the group is subordinate to--or a child of--another group. The *has-as-a-member* relationship indicates that the group is a parent of another group. By creating group-to-group relationships, the user is given access not only to objects defined to the groups to which the user has direct access, but to objects defined to subgroups as well. The *group-to-group* relationship has no relationship attributes, but it does have audit attributes.

Just as you can migrate objects, you can migrate groups (see [Group is-a-root-of Migration](#)).

Group is-a-root-of Migration



When you create a group in one repository, you can migrate the group entity to other repositories rather than create a new group entity in each repository. When you create a relationship between a group and a migration entity, that group becomes part of a migration package. You can migrate only the group or the group with related users and projects by specifying scope when you define the relationship attribute. [Is-a-root-of Migration Relationship Attribute](#) describes the attribute for the *is-a-root-of* Migration relationship.

Is-a-root-of Migration Relationship Attribute		
Attribute	Requirements	Description

Scope Type	Required	Either "Entity Only", "One level", or "Entity hierarchy" from the following: <ul style="list-style-type: none"> Entity only One level Prepare hierarchy Entity hierarchy Drawing
------------	----------	---

Project Entity

Use the Project entity to group sets of objects. Thus, a project is a subset of the objects in a repository. Moreover, a project contains a subset of objects of particular types. An ADM project, for example, contains only objects of the Administration Information Model. Whenever you add an entity, you must assign values for several attributes. Each project entity has two types of attributes: project and audit. [Project Entity Attributes](#) describes each project entity attribute.

Unless specified otherwise, blanks are not allowed.

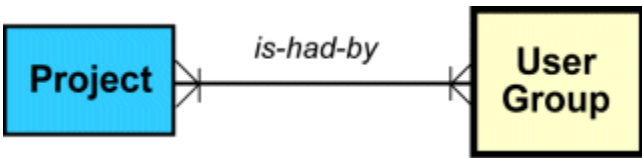
Project Entity Attributes		
Attribute	Requirements	Description
Project Name	Required	A unique alphanumeric string up to thirty characters that identifies the project.
Project ID	Required	A unique alphanumeric string up to four characters that describes the purpose of the group. No blanks are allowed. The project ID is displayed in the Description column of the project list produced by the PROJECTS command.

An instance of a project entity can have the following relationships:

- Is-had-by* Group
- Is-a-root-of* Migration

The projects to which a user is related--~~through one or more group relationships~~--determine the objects the user can access (see [Project is-had-by Group](#)).

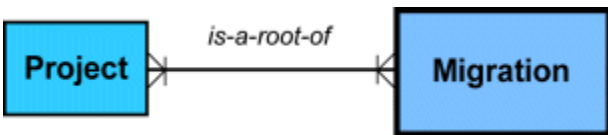
Project is-had-by Group



Project *is-had-by* Group is just another perspective of the group has project relationship. See [Has project Relationship Attributes](#) for a description of the relationship attributes.

Just as you can migrate objects, you can migrate projects (see [Project is-a-root-of Migration](#)).

Project is-a-root-of Migration



When you create a project in one repository, you can migrate the project entity to other repositories rather than create a new project entity in each repository. When you create a relationship between a project and a migration entity, that project becomes part of a migration package. You specify that only the entity is to be migrated when you define the relationship attribute. [Is-a-root-of Migration Relationship Attribute](#) describes the attribute for the *is-a-root-of* Migration relationship.

Is-a-root-of Migration Relationship Attribute		
Attribute	Requirements	Description
Scope Type	Required	"Entity Only" from the following: <ul style="list-style-type: none"> • Entity only • One level • Prepare hierarchy • Entity hierarchy • Drawing

Repository Entity

The repository entity is an object that refers to an entire repository---local, group, or mainframe. Repository entities always reside in version A of an enterprise repository. An administrator must be authorized to version A in order to create a repository entity.

	The repository entity defines whether or not a repository can export or import data. You cannot migrate without defining the repository entity.
--	---

Each repository entity has two types of attributes: repository and audit. [Repository Entity Attributes](#) describes each repository entity attribute.

	Unless specified otherwise, blanks are not allowed.
--	---

Repository Entity Attributes		
Attribute	Requirements	Description
Name	Required	A unique alphanumeric string up to thirty characters that identifies the repository.
Description	Optional	A unique alphanumeric string up to thirty characters, including blanks, that describes the purpose of the repository.
Type	Required	One of the following: <ul style="list-style-type: none"> • Workgroup Repository • Personal (local) Repository • Enterprise (mainframe) Repository
Implementation Name	Required for MRE only	A unique alphanumeric string up to eight characters that refers to an INI file with information about the mainframe repository. The file is used during migration. For further information, see <i>Enterprise Migration Guide</i> .

An instance of a repository entity can have the following relationships:

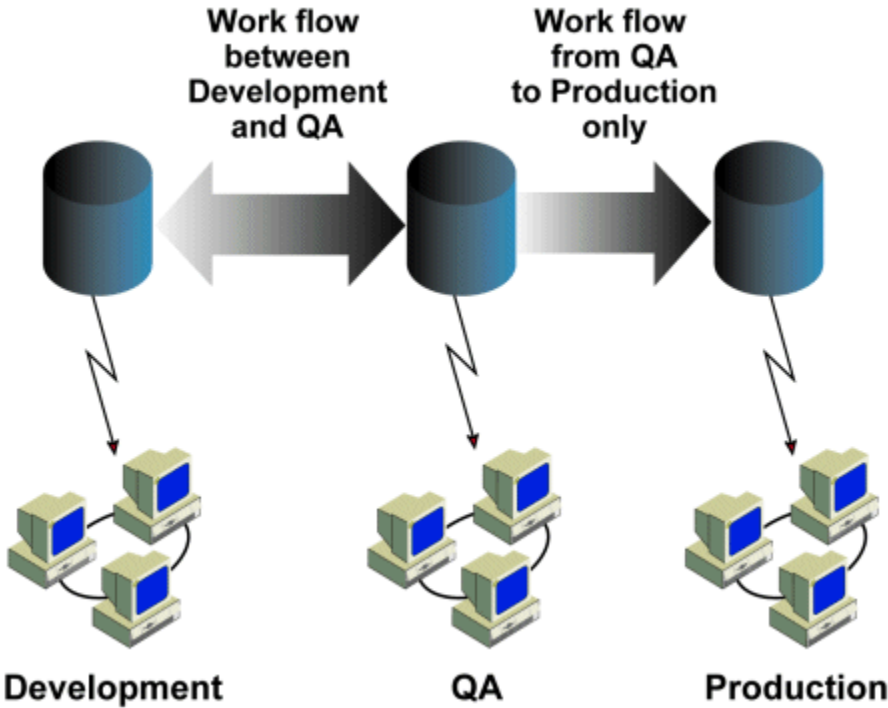
- *Can-import-data-from* Repository
- *Can-migrate-data-to* Repository

Repository can-migrate-data-to Repository



By relating repositories, the administrator defines the flow of work within a particular configuration. For example, you can determine that work can move from the development repository to the quality assurance (QA) repository, and from the QA repository back to the development repository. However, you can define the direction of work flow from the QA repository to the production repository to be *can-export-to* Repository only. See [Repository Work Flow](#).

Repository Work Flow



[Has repository Relationship Attributes](#) describes the "Has repository" relationship attributes.

Has repository Relationship Attributes		
Attribute	Requirements	Description
Direction	Required	One of the following choices to indicate the direction of workflow: <ul style="list-style-type: none"> • <i>Can-export-to</i> repository • <i>Can-import-from</i> repository
Source Version	Required	
Target Version	Required	

Enterprise Repository Objects

Target Version	Required	
----------------	----------	--

Enterprise Repository Objects

The Enterprise Repository contains the objects needed to build applications. This chapter provides a complete list of the objects, methods, and actions you can use with AppBuilder.

Using the Enterprise Repository Objects

The following tables show all the objects—entities, relationships, and types—in an enterprise repository and the methods or actions that can be applied to each object. When you add a group-to-project relationship to an enterprise repository, you grant users in the group access to objects in the project. In doing so, you must specify:

- The *scope* of user access—what types of objects the group members can access.

To specify a type of object, use the name in the *Object type* column of the following table.

- The *method* name—what methods group members can perform on objects.

To specify a method, use the name in the *Method name* column of the following table.

For information about granting users access to objects and methods, see the *Enterprise Administration Guide* .

Listing of Enterprise Repository Objects, Methods, and Actions

[See A](#) / [See B](#) / [See C](#) / [See D](#) / [See E](#) / [See F](#) / [See G](#) / [See H](#) / [See I](#) / [See K](#) / [See L](#) / [See M](#) / [See O](#) / [See P](#) / [See R](#) / [See S](#) / [See T](#) / [See U](#) / [See V](#) / [See W](#)

Enterprise Repository Objects			
<i>Object type</i>	<i>User action</i>	<i>Method name</i>	<i>Description</i>
A			
ADMIN	ACTIONS	ACTIONS	List Entity Actions
ADMIN	MENU	SECADM	Security Administration
APPCFG	ACTIONS	ACTIONS	List Entity Actions
APPCFG	ADDE	ADDE	Add an Entity
APPCFG	ADDR	ADDR	Add a Relation
APPCFG	BE	BE	Browse Entity Attributes
APPCFG	BK	HMFKEYB	Browse Keywords
APPCFG	BR	BR	Browse Relation Attributes
APPCFG	BTXE	TEXTBE	Browse Text for Entity
APPCFG	BTXR	TEXTBR	Browse Text for Relation
APPCFG	CO	HMFCOS0	Change Ownership
APPCFG	DE	DELETEE	Delete an Entity
APPCFG	DR	DELETER	Delete a Relation
APPCFG	EW	HMFYWU	Extended Where Used Report
APPCFG	K	HMFKEY0	Define Keywords
APPCFG	LE	LISTE	List Entity Instances
APPCFG	LOCK	HMFUOLK	Lock an Entity
APPCFG	LR	LISTR	List Relation Instances
APPCFG	ME	MAINTE	Maintain Entity Attributes
APPCFG	MR	MAINTR	Maintain Relation Attributes
APPCFG	REFRESH	REFRESH	Refresh Entity
APPCFG	RES	RESGLBL	View Results of Actions
APPCFG	SA	SAVEAS	Save As Current Entity
APPCFG	SRCH	SEARCH	Search for Keywords
APPCFG	TXE	TEXTE	Define Text for Entity
APPCFG	TXR	TEXTR	Define Text for Relation
APPCFG	UEMTH	UEMTH	HPS User Entity Method
APPCFG	UNLOCK	HMFUOFR	Unlock an Entity
APPCFG	URMTH	URMTH	HPS User Relation Method

ATTRIB	ACTIONS	ACTIONS	List Entity Actions
ATTRIB	ADDE	ADDE	Add an Entity
ATTRIB	ADDR	ADDR	Add a Relation
ATTRIB	BE	BE	Browse Entity Attributes
ATTRIB	BK	HMFKEYB	Browse Keywords
ATTRIB	BR	BR	Browse Relation Attributes
ATTRIB	BTXE	TEXTBE	Browse Text for Entity
ATTRIB	BTXR	TEXTBR	Browse Text for Relation
ATTRIB	CO	HMFOS0	Change Ownership
ATTRIB	DE	DELETEE	Delete an Entity
ATTRIB	DR	DELETER	Delete a Relation
ATTRIB	EW	HMFEXWU	Extended Where Used Report
ATTRIB	K	HMFKEY0	Define Keywords
ATTRIB	LE	LISTE	List Entity Instances
ATTRIB	LOCK	HMFUOLK	Lock an Entity
ATTRIB	LR	LISTR	List Relation Instances
ATTRIB	ME	MAINTE	Maintain Entity Attributes
ATTRIB	MR	MAINTR	Maintain Relation Attributes
ATTRIB	REFRESH	REFRESH	Refresh Entity
ATTRIB	RES	RESGLBL	View Results of Actions
ATTRIB	SA	SAVEAS	Save As Current Entity
ATTRIB	SRCH	SEARCH	Search for Keywords
ATTRIB	TXE	TEXTE	Define Text for Entity
ATTRIB	TXR	TEXTR	Define Text for Relation
ATTRIB	UEMTH	UEMTH	HPS User Entity Method
ATTRIB	UNLOCK	HMFUOFR	Unlock an Entity
ATTRIB	URMTH	URMTH	HPS User Relation Method
B			
BITMAP	ACTIONS	ACTIONS	List Entity Actions
BITMAP	ADDE	ADDE	Add an Entity
BITMAP	ADDR	ADDR	Add a Relation
BITMAP	BE	BE	Browse Entity Attributes
BITMAP	BK	HMFKEYB	Browse Keywords
BITMAP	BR	BR	Browse Relation Attributes
BITMAP	BTXE	TEXTBE	Browse Text for Entity
BITMAP	BTXR	TEXTBR	Browse Text for Relation
BITMAP	CO	HMFOS0	Change Ownership
BITMAP	DE	DELETEE	Delete an Entity
BITMAP	DR	DELETER	Delete a Relation

BITMAP	EW	HMFXTWU	Extended Where Used Report
BITMAP	K	HMFKEY0	Define Keywords
BITMAP	LE	LISTE	List Entity Instances
BITMAP	LOCK	HMFUOLK	Lock an Entity
BITMAP	LR	LISTR	List Relation Instances
BITMAP	ME	MAINTE	Maintain Entity Attributes
BITMAP	MR	MAINTR	Maintain Relation Attributes
BITMAP	REFRESH	REFRESH	Refresh Entity
BITMAP	RES	RESGLBL	View Results of Actions
BITMAP	SA	SAVEA	Save As Current Entity
BITMAP	SRCH	SEARCH	Search for Keywords
BITMAP	TXE	TEXTE	Define Text for Entity
BITMAP	TXR	TEXTR	Define Text for Relation
BITMAP	UEMTH	UEMTH	HPS User Entity Method
BITMAP	UNLOCK	HMFUOFR	Unlock an Entity
BITMAP	URMTH	URMTH	HPS User Relation Method
BMPIMP	ACTIONS	ACTIONS	List Entity Actions
BMPIMP	ADDE	ADDE	Add an Entity
BMPIMP	ADDR	ADDR	Add a Relation
BMPIMP	BE	BE	Browse Entity Attributes
BMPIMP	BK	HMFKEYB	Browse Keywords
BMPIMP	BR	BR	Browse Relation Attributes
BMPIMP	BTXE	TEXTBE	Browse Text for Entity
BMPIMP	BTXR	TEXTBR	Browse Text for Relation
BMPIMP	CO	HMFOS0	Change Ownership
BMPIMP	DE	DELETEE	Delete an Entity
BMPIMP	DR	DELETER	Delete a Relation
BMPIMP	EW	HMFXTWU	Extended Where Used Report
BMPIMP	K	HMFKEY0	Define Keywords
BMPIMP	LE	LISTE	List Entity Instances
BMPIMP	LOCK	HMFUOLK	Lock an Entity
BMPIMP	LR	LISTR	List Relation Instances
BMPIMP	ME	MAINTE	Maintain Entity Attributes
BMPIMP	MR	MAINTR	Maintain Relation Attributes
BMPIMP	REFRESH	REFRESH	Refresh Entity
BMPIMP	RES	RESGLBL	View Results of Actions
BMPIMP	SA	SAVEAS	Save As Current Entity
BMPIMP	SRCH	SEARCH	Search for Keywords
BMPIMP	TXE	TEXTE	Define Text for Entity

BMPIMP	TXR	TEXTR	Define Text for Relation
BMPIMP	UEMTH	UEMTH	HPS User Entity Method
BMPIMP	UNLOCK	HMFUOFR	Unlock an Entity
BMPIMP	URMTH	URMTH	HPS User Relation Method
BUSOBJ	ACTIONS	ACTIONS	List Entity Actions
BUSOBJ	ADDE	ADDE	Add an Entity
BUSOBJ	ADDR	ADDR	Add a Relation
BUSOBJ	BE	BE	Browse Entity Attributes
BUSOBJ	BK	HFMFMKEYB	Browse Keywords
BUSOBJ	BR	BR	Browse Relation Attributes
BUSOBJ	BTXE	TEXTBE	Browse Text for Entity
BUSOBJ	BTXR	TEXTBR	Browse Text for Relation
BUSOBJ	CO	HMFCOS0	Change Ownership
BUSOBJ	DE	DELETEE	Delete an Entity
BUSOBJ	DR	DELETER	Delete a Relation
BUSOBJ	EW	HMFXTWU	Extended Where Used Report
BUSOBJ	K	HFMFMKEY0	Define Keywords
BUSOBJ	LE	LISTE	List Entity Instances
BUSOBJ	LOCK	HMFUOLK	Lock an Entity
BUSOBJ	LR	LISTR	List Relation Instances
BUSOBJ	ME	MAINTE	Maintain Entity Attributes
BUSOBJ	MR	MAINTR	Maintain Relation Attributes
BUSOBJ	REFRESH	REFRESH	Refresh Entity
BUSOBJ	RES	RESGLBL	View Results of Actions
BUSOBJ	SA	SAVEAS	Save As Current Entity
BUSOBJ	SRCH	SEARCH	Search for Keywords
BUSOBJ	TXE	TEXTE	Define Text for Entity
BUSOBJ	TXR	TEXTR	Define Text for Relation
BUSOBJ	UEMTH	UEMTH	HPS User Entity Method
BUSOBJ	UNLOCK	HMFUOFR	Unlock an Entity
BUSOBJ	URMTH	URMTH	HPS User Relation Method
C			
CELL	ACTIONS	ACTIONS	List Entity Actions
CELL	ADDE	ADDE	Add an Entity
CELL	ADDR	ADDR	Add a Relation
CELL	BE	BE	Browse Entity Attributes
CELL	BK	HFMFMKEYB	Browse Keywords
CELL	BR	BR	Browse Relation Attributes
CELL	BTXE	TEXTBE	Browse Text for Entity

CELL	BTXR	TEXTBR	Browse Text for Relation
CELL	CO	HMF COS0	Change Ownership
CELL	DE	DELETEE	Delete an Entity
CELL	DR	DELETER	Delete a Relation
CELL	EW	HMF XWU	Extended Where Used Report
CELL	K	HMF MKEY0	Define Keywords
CELL	LE	LISTE	List Entity Instances
CELL	LOCK	HMF UOLK	Lock an Entity
CELL	LR	LISTR	List Relation Instances
CELL	ME	MAINTE	Maintain Entity Attributes
CELL	MR	MAINTR	Maintain Relation Attributes
CELL	REFRESH	REFRESH	Refresh Entity
CELL	RES	RESGLBL	View Results of Actions
CELL	SA	SAVEAS	Save As Current Entity
CELL	SRCH	SEARCH	Search for Keywords
CELL	TXE	TEXTE	Define Text for Entity
CELL	TXR	TEXTR	Define Text for Relation
CELL	UEMTH	UEMTH	HPS User Entity Method
CELL	UNLOCK	HMF UOFR	Unlock an Entity
CELL	URMTH	URMTH	HPS User Relation Method
COLLECT	ACTIONS	ACTIONS	List Entity Actions
COLLECT	ADDE	ADDE	Add an Entity
COLLECT	ADDR	ADDR	Add a Relation
COLLECT	BE	BE	Browse Entity Attributes
COLLECT	BK	HMF MKEYB	Browse Keywords
COLLECT	BR	BR	Browse Relation Attributes
COLLECT	BTXE	TEXTBE	Browse Text for Entity
COLLECT	BTXR	TEXTBR	Browse Text for Relation
COLLECT	CO	HMF COS0	Change Ownership
COLLECT	DE	DELETEE	Delete an Entity
COLLECT	DR	DELETER	Delete a Relation
COLLECT	EW	HMF XWU	Extended Where Used Report
COLLECT	K	HMF MKEY0	Define Keywords
COLLECT	LE	LISTE	List Entity Instances
COLLECT	LOCK	HMF UOLK	Lock an Entity
COLLECT	LR	LISTR	List Relation Instances
COLLECT	ME	MAINTE	Maintain Entity Attributes
COLLECT	MR	MAINTR	Maintain Relation Attributes
COLLECT	REFRESH	REFRESH	Refresh Entity

COLLECT	RES	RESGLBL	View Results of Actions
COLLECT	SA	SAVEAS	Save As Current Entity
COLLECT	SRCH	SEARCH	Search for Keywords
COLLECT	TXE	TEXTE	Define Text for Entity
COLLECT	TXR	TEXTR	Define Text for Relation
COLLECT	UEMTH	UEMTH	HPS User Entity Method
COLLECT	UNLOCK	HMFUOFR	Unlock an Entity
COLLECT	URMTH	URMTH	HPS User Relation Method
COLUMN	ACTIONS	ACTIONS	List Entity Actions
COLUMN	ADDE	ADDE	Add an Entity
COLUMN	ADDR	ADDR	Add a Relation
COLUMN	BE	BE	Browse Entity Attributes
COLUMN	BK	HFMFKEYB	Browse Keywords
COLUMN	BR	BR	Browse Relation Attributes
COLUMN	BTXE	TEXTBE	Browse Text for Entity
COLUMN	BTXR	TEXTBR	Browse Text for Relation
COLUMN	CO	HMFCOS0	Change Ownership
COLUMN	DE	DELETEE	Delete an Entity
COLUMN	DR	DELETER	Delete a Relation
COLUMN	EW	HMFYWU	Extended Where Used Report
COLUMN	K	HFMFKEY0	Define Keywords
COLUMN	LE	LISTE	List Entity Instances
COLUMN	LOCK	HMFUOLK	Lock an Entity
COLUMN	LR	LISTR	List Relation Instances
COLUMN	ME	MAINTE	Maintain Entity Attributes
COLUMN	MR	MAINTR	Maintain Relation Attributes
COLUMN	REFRESH	REFRESH	Refresh Entity
COLUMN	RES	RESGLBL	View Results of Actions
COLUMN	SA	SAVEAS	Save As Current Entity
COLUMN	SRCH	SEARCH	Search for Keywords
COLUMN	TXE	TEXTE	Define Text for Entity
COLUMN	TXR	TEXTR	Define Text for Relation
COLUMN	UEMTH	UEMTH	HPS User Entity Method
COLUMN	UNLOCK	HMFUOFR	Unlock an Entity
COLUMN	URMTH	URMTH	HPS User Relation Method
COMP	ACTIONS	ACTIONS	List Entity Actions
COMP	ADDE	ADDE	Add an Entity
COMP	ADDR	ADDR	Add a Relation
COMP	ASSIGNCU	HMFACU0	Assign Configuration Unit

COMP	B	HMFMEDB	Browse Source Code
COMP	BE	BE	Browse Entity Attributes
COMP	BINDPKG	BINDPKG	Bind DB2 Package
COMP	BK	HFMKEYB	Browse Keywords
COMP	BR	BR	Browse Relation Attributes
COMP	BTXE	TEXTBE	Browse Text for Entity
COMP	BTXR	TEXTBR	Browse Text for Relation
COMP	CO	HMFCOS0	Change Ownership
COMP	DE	DELETEE	Delete an Entity
COMP	DR	DELETER	Delete a Relation
COMP	EW	HMFYWU	Extended Where Used Report
COMP	K	HFMKEY0	Define Keywords
COMP	LE	LISTE	List Entity Instances
COMP	LISTRBD	LISTRBD	List Rebuild Contents
COMP	LOCK	HMFUOLK	Lock an Entity
COMP	LR	LISTR	List Relation Instances
COMP	ME	MAINTE	Maintain Entity Attributes
COMP	MR	MAINTR	Maintain Relation Attributes
COMP	PR	PREPCOMP	Prepare Component
COMP	RB	HMFMBND	ReBind DB2 Plans
COMP	REFRESH	REFRESH	Refresh Entity
COMP	REP	HMFREP	Produce Report
COMP	RES	RESCOMP	View Results of Comp Actions
COMP	RIN	HMFMLU2	Re-Install into CICS
COMP	S	HFMEDS	Define Source Code
COMP	SA	SAVEAS	Save As Current Entity
COMP	SRCH	SEARCH	Search for Keywords
COMP	SUPERPR	SUPERPR	Super Prepare
COMP	TXE	TEXTE	Define Text for Entity
COMP	TXR	TEXTR	Define Text for Relation
COMP	UEMTH	UEMTH	HPS User Entity Method
COMP	UNLOCK	HMFUOFR	Unlock an Entity
COMP	URMTH	URMTH	HPS User Relation Method
CURSYS	ACTIONS	ACTIONS	List Entity Actions
CURSYS	ADDE	ADDE	Add an Entity
CURSYS	ADDR	ADDR	Add a Relation
CURSYS	BE	BE	Browse Entity Attributes
CURSYS	BK	HFMKEYB	Browse Keywords
CURSYS	BR	BR	Browse Relation Attributes

CURSYS	BTXE	TEXTBE	Browse Text for Entity
CURSYS	BTXR	TEXTBR	Browse Text for Relation
CURSYS	CO	HMF COS0	Change Ownership
CURSYS	DE	DELETEE	Delete an Entity
CURSYS	DR	DELETER	Delete a Relation
CURSYS	EW	HMF XWU	Extended Where Used Report
CURSYS	K	HMF MKEY0	Define Keywords
CURSYS	LE	LISTE	List Entity Instances
CURSYS	LOCK	HMF UOLK	Lock an Entity
CURSYS	LR	LISTR	List Relation Instances
CURSYS	ME	MAINTE	Maintain Entity Attributes
CURSYS	MR	MAINTR	Maintain Relation Attributes
CURSYS	REFRESH	REFRESH	Refresh Entity
CURSYS	RES	RESGLBL	View Results of Actions
CURSYS	SA	SAVEAS	Save As Current Entity
CURSYS	SRCH	SEARCH	Search for Keywords
CURSYS	TXE	TEXTE	Define Text for Entity
CURSYS	TXR	TEXTR	Define Text for Relation
CURSYS	UEMTH	UEMTH	HPS User Entity Method
CURSYS	UNLOCK	HMF UOFR	Unlock an Entity
CURSYS	URMTH	URMTH	HPS User Relation Method
D			
DATA	ACTIONS	ACTIONS	List Entity Actions
DATA	ADDE	ADDE	Add an Entity
DATA	ADDR	ADDR	Add a Relation
DATA	BE	BE	Browse Entity Attributes
DATA	BK	HMF MKEYB	Browse Keywords
DATA	BR	BR	Browse Relation Attributes
DATA	BTXE	TEXTBE	Browse Text for Entity
DATA	BTXR	TEXTBR	Browse Text for Relation
DATA	CO	HMF COS0	Change Ownership
DATA	DE	DELETEE	Delete an Entity
DATA	DR	DELETER	Delete a Relation
DATA	EW	HMF XWU	Extended Where Used Report
DATA	K	HMF MKEY0	Define Keywords
DATA	LE	LISTE	List Entity Instances
DATA	LOCK	HMF UOLK	Lock an Entity
DATA	LR	LISTR	List Relation Instances
DATA	ME	MAINTE	Maintain Entity Attributes

DATA	MR	MAINTR	Maintain Relation Attributes
DATA	REFRESH	REFRESH	Refresh Entity
DATA	RES	RESGLBL	View Results of Actions
DATA	SA	SAVEAS	Save As Current Entity
DATA	SRCH	SEARCH	Search for Keywords
DATA	TXE	TEXTE	Define Text for Entity
DATA	TXR	TEXTR	Define Text for Relation
DATA	UEMTH	UEMTH	HPS User Entity Method
DATA	UNLOCK	HMFUOFR	Unlock an Entity
DATA	URMTH	URMTH	HPS User Relation Method
DATABAS	ACTIONS	ACTIONS	List Entity Actions
DATABAS	ADDE	ADDE	Add an Entity
DATABAS	ADDR	ADDR	Add a Relation
DATABAS	B	HFMEDB	Browse Source Code
DATABAS	BE	BE	Browse Entity Attributes
DATABAS	BK	HFMKEYB	Browse Keywords
DATABAS	BR	BR	Browse Relation Attributes
DATABAS	BTXE	TEXTBE	Browse Text for Entity
DATABAS	BTXR	TEXTBR	Browse Text for Relation
DATABAS	CO	HMFCOS0	Change Ownership
DATABAS	DE	DELETEE	Delete an Entity
DATABAS	DR	DELETER	Delete a Relation
DATABAS	EW	HMFYWU	Extended Where Used Report
DATABAS	K	HFMKEY0	Define Keywords
DATABAS	LE	LISTE	List Entity Instances
DATABAS	LOCK	HMFUOLK	Lock an Entity
DATABAS	LR	LISTR	List Relation Instances
DATABAS	ME	MAINTE	Maintain Entity Attributes
DATABAS	MR	MAINTR	Maintain Relation Attributes
DATABAS	REFRESH	REFRESH	Refresh Entity
DATABAS	RES	RESGLBL	View Results of Actions
DATABAS	S	HFMEDS	Define Source Code
DATABAS	SA	SAVEAS	Save As Current Entity
DATABAS	SRCH	SEARCH	Search for Keywords
DATABAS	TXE	TEXTE	Define Text for Entity
DATABAS	TXR	TEXTR	Define Text for Relation
DATABAS	UEMTH	UEMTH	HPS User Entity Method
DATABAS	UNLOCK	HMFUOFR	Unlock an Entity
DATABAS	URMTH	URMTH	HPS User Relation Method

DRAWNG	ACTIONS	ACTIONS	List Entity Actions
DRAWNG	ADDE	ADDE	Add an Entity
DRAWNG	ADDR	ADDR	Add a Relation
DRAWNG	BE	BE	Browse Entity Attributes
DRAWNG	BK	HMFKEYB	Browse Keywords
DRAWNG	BR	BR	Browse Relation Attributes
DRAWNG	BTXE	TEXTBE	Browse Text for Entity
DRAWNG	BTXR	TEXTBR	Browse Text for Relation
DRAWNG	CO	HMFOS0	Change Ownership
DRAWNG	DE	DELETEE	Delete an Entity
DRAWNG	DR	DELETER	Delete a Relation
DRAWNG	EW	HMFYWU	Extended Where Used Report
DRAWNG	K	HMFKEY0	Define Keywords
DRAWNG	LE	LISTE	List Entity Instances
DRAWNG	LOCK	HMFUOLK	Lock an Entity
DRAWNG	LR	LISTR	List Relation Instances
DRAWNG	ME	MAINTE	Maintain Entity Attributes
DRAWNG	MR	MAINTR	Maintain Relation Attributes
DRAWNG	REFRESH	REFRESH	Refresh Entity
DRAWNG	RES	RESGLBL	View Results of Actions
DRAWNG	SA	SAVEAS	Save As Current Entity
DRAWNG	SRCH	SEARCH	Search for Keywords
DRAWNG	TXE	TEXTE	Define Text for Entity
DRAWNG	TXR	TEXTR	Define Text for Relation
DRAWNG	UEMTH	UEMTH	HPS User Entity Method
DRAWNG	UNLOCK	HMFUOFR	Unlock an Entity
DRAWNG	URMTH	URMTH	HPS User Relation Method
E			
ENT	ACTIONS	ACTIONS	List Entity Actions
ENT	ADDE	ADDE	Add an Entity
ENT	ADDR	ADDR	Add a Relation
ENT	BE	BE	Browse Entity Attributes
ENT	BK	HMFKEYB	Browse Keywords
ENT	BR	BR	Browse Relation Attributes
ENT	BTXE	TEXTBE	Browse Text for Entity
ENT	BTXR	TEXTBR	Browse Text for Relation
ENT	CO	HMFOS0	Change Ownership
ENT	DE	DELETEE	Delete an Entity
ENT	DR	DELETER	Delete a Relation

ENT	EW	HMFXTWU	Extended Where Used Report
ENT	K	HMFMTKEY0	Define Keywords
ENT	LE	LISTE	List Entity Instances
ENT	LOCK	HMFUOLK	Lock an Entity
ENT	LR	LISTR	List Relation Instances
ENT	ME	MAINTE	Maintain Entity Attributes
ENT	MR	MAINTR	Maintain Relation Attributes
ENT	REFRESH	REFRESH	Refresh Entity
ENT	RES	RESGLBL	View Results of Actions
ENT	SA	SAVEAS	Save As Current Entity
ENT	SRCH	SEARCH	Search for Keywords
ENT	TXE	TEXTE	Define Text for Entity
ENT	TXR	TEXTR	Define Text for Relation
ENT	UEMTH	UEMTH	HPS User Entity Method
ENT	UNLOCK	HMFUOFR	Unlock an Entity
ENT	URMTH	URMTH	HPS User Relation Method
EVENT	ACTIONS	ACTIONS	List Entity Actions
EVENT	ADDE	ADDE	Add an Entity
EVENT	ADDR	ADDR	Add a Relation
EVENT	BE	BE	Browse Entity Attributes
EVENT	BK	HMFMTKEYB	Browse Keywords
EVENT	BR	BR	Browse Relation Attributes
EVENT	BTXE	TEXTBE	Browse Text for Entity
EVENT	BTXR	TEXTBR	Browse Text for Relation
EVENT	CO	HMFOS0	Change Ownership
EVENT	DE	DELETEE	Delete an Entity
EVENT	DR	DELETER	Delete a Relation
EVENT	EW	HMFXTWU	Extended Where Used Report
EVENT	K	HMFMTKEY0	Define Keywords
EVENT	LE	LISTE	List Entity Instances
EVENT	LOCK	HMFUOLK	Lock an Entity
EVENT	LR	LISTR	List Relation Instances
EVENT	ME	MAINTE	Maintain Entity Attributes
EVENT	MR	MAINTR	Maintain Relation Attributes
EVENT	REFRESH	REFRESH	Refresh Entity
EVENT	RES	RESGLBL	View Results of Actions
EVENT	SA	SAVEAS	Save As Current Entity
EVENT	SRCH	SEARCH	Search for Keywords
EVENT	TXE	TEXTE	Define Text for Entity

EVENT	TXR	TEXTR	Define Text for Relation
EVENT	UEMTH	UEMTH	HPS User Entity Method
EVENT	UNLOCK	HMFUOFR	Unlock an Entity
EVENT	URMTH	URMTH	HPS User Relation Method
F			
FILE	ACTIONS	ACTIONS	List Entity Actions
FILE	ADDE	ADDE	Add an Entity
FILE	ADDR	ADDR	Add a Relation
FILE	ASSIGNCU	HMFACU0	Assign Configuration Unit
FILE	BE	BE	Browse Entity Attributes
FILE	BK	HMFKEYB	Browse Keywords
FILE	BR	BR	Browse Relation Attributes
FILE	BTXE	TEXTBE	Browse Text for Entity
FILE	BTXR	TEXTBR	Browse Text for Relation
FILE	CO	HMFCOS0	Change Ownership
FILE	DE	DELETEE	Delete an Entity
FILE	DR	DELETER	Delete a Relation
FILE	EW	HMFEXWU	Extended Where Used Report
FILE	GD	HMFMDL	Generate DDL Statements
FILE	K	HMFKEY0	Define Keywords
FILE	LE	LISTE	List Entity Instances
FILE	LISTRBD	LISTRBD	List Rebuild Contents
FILE	LOCK	HMFUOLK	Lock an Entity
FILE	LR	LISTR	List Relation Instances
FILE	ME	MAINTE	Maintain Entity Attributes
FILE	MR	MAINTR	Maintain Relation Attributes
FILE	PR	PREPFILE	Prepare File
FILE	REFRESH	REFRESH	Refresh Entity
FILE	REP	HMFREP	Produce Report
FILE	RES	RESFILE	View Results of File Actions
FILE	SA	SAVEAS	Save As Current Entity
FILE	SRCH	SEARCH	Search for Keywords
FILE	SUPERPR	SUPERPR	Super Prepare
FILE	TXE	TEXTE	Define Text for Entity
FILE	TXR	TEXTR	Define Text for Relation
FILE	UEMTH	UEMTH	HPS User Entity Method
FILE	UNLOCK	HMFUOFR	Unlock an Entity
FILE	URMTH	URMTH	HPS User Relation Method
FLD	ACTIONS	ACTIONS	List Entity Actions

FLD	ADDE	ADDE	Add an Entity
FLD	ADDR	ADDR	Add a Relation
FLD	BE	BE	Browse Entity Attributes
FLD	BK	HMFKEYB	Browse Keywords
FLD	BR	BR	Browse Relation Attributes
FLD	BTXE	TEXTBE	Browse Text for Entity
FLD	BTXR	TEXTBR	Browse Text for Relation
FLD	CO	HMFOS0	Change Ownership
FLD	DE	DELETEE	Delete an Entity
FLD	DR	DELETER	Delete a Relation
FLD	EW	HMFOWU	Extended Where Used Report
FLD	K	HMFKEY0	Define Keywords
FLD	LE	LISTE	List Entity Instances
FLD	LOCK	HMFOLK	Lock an Entity
FLD	LR	LISTR	List Relation Instances
FLD	ME	MAINTE	Maintain Entity Attributes
FLD	MR	MAINTR	Maintain Relation Attributes
FLD	REFRESH	REFRESH	Refresh Entity
FLD	RES	RESGLBL	View Results of Actions
FLD	SA	SAVEAS	Save As Current Entity
FLD	SRCH	SEARCH	Search for Keywords
FLD	TXE	TEXTE	Define Text for Entity
FLD	TXR	TEXTR	Define Text for Relation
FLD	UEMTH	UEMTH	HPS User Entity Method
FLD	UNLOCK	HMFUFR	Unlock an Entity
FLD	URMTH	URMTH	HPS User Relation Method
FORM	ACTIONS	ACTIONS	List Entity Actions
FORM	ADDE	ADDE	Add an Entity
FORM	ADDR	ADDR	Add a Relation
FORM	B	HFMEDB	Browse Source Code
FORM	BE	BE	Browse Entity Attributes
FORM	BK	HMFKEYB	Browse Keywords
FORM	BR	BR	Browse Relation Attributes
FORM	BTXE	TEXTBE	Browse Text for Entity
FORM	BTXR	TEXTBR	Browse Text for Relation
FORM	CO	HMFOS0	Change Ownership
FORM	DE	DELETEE	Delete an Entity
FORM	DR	DELETER	Delete a Relation
FORM	EW	HMFOWU	Extended Where Used Report

FORM	K	HMFKEY0	Define Keywords
FORM	LE	LISTE	List Entity Instances
FORM	LOCK	HMFUOLK	Lock an Entity
FORM	LR	LISTR	List Relation Instances
FORM	ME	MAINTE	Maintain Entity Attributes
FORM	MR	MAINTR	Maintain Relation Attributes
FORM	REFRESH	REFRESH	Refresh Entity
FORM	RES	RESGLBL	View Results of Actions
FORM	S	HFMEDS	Define Source Code
FORM	SA	SAVEAS	Save As Current Entity
FORM	SRCH	SEARCH	Search for Keywords
FORM	TXE	TEXTE	Define Text for Entity
FORM	TXR	TEXTR	Define Text for Relation
FORM	UEMTH	UEMTH	HPS User Entity Method
FORM	UNLOCK	HMFUOFR	Unlock an Entity
FORM	URMTH	URMTH	HPS User Relation Method
FSCHEME	ACTIONS	ACTIONS	List Entity Actions
FSCHEME	ADDE	ADDE	Add an Entity
FSCHEME	ADDR	ADDR	Add a Relation
FSCHEME	BE	BE	Browse Entity Attributes
FSCHEME	BK	HMFKEYB	Browse Keywords
FSCHEME	BR	BR	Browse Relation Attributes
FSCHEME	BTXE	TEXTBE	Browse Text for Entity
FSCHEME	BTXR	TEXTBR	Browse Text for Relation
FSCHEME	CO	HMFCOS0	Change Ownership
FSCHEME	DE	DELETEE	Delete an Entity
FSCHEME	DR	DELETER	Delete a Relation
FSCHEME	EW	HMFYWU	Extended Where Used Report
FSCHEME	K	HMFKEY0	Define Keywords
FSCHEME	LE	LISTE	List Entity Instances
FSCHEME	LOCK	HMFUOLK	Lock an Entity
FSCHEME	LR	LISTR	List Relation Instances
FSCHEME	ME	MAINTE	Maintain Entity Attributes
FSCHEME	MR	MAINTR	Maintain Relation Attributes
FSCHEME	REFRESH	REFRESH	Refresh Entity
FSCHEME	RES	RESGLBL	View Results of Actions
FSCHEME	SA	SAVEAS	Save As Current Entity
FSCHEME	SRCH	SEARCH	Search for Keywords
FSCHEME	TXE	TEXTE	Define Text for Entity

FSCHEME	TXR	TEXTR	Define Text for Relation
FSCHEME	UEMTH	UEMTH	HPS User Entity Method
FSCHEME	UNLOCK	HMFUOFR	Unlock an Entity
FSCHEME	URMTH	URMTH	HPS User Relation Method
FUNC	ACTIONS	ACTIONS	List Entity Actions
FUNC	ADDE	ADDE	Add an Entity
FUNC	ADDR	ADDR	Add a Relation
FUNC	BE	BE	Browse Entity Attributes
FUNC	BK	HMFKEYB	Browse Keywords
FUNC	BR	BR	Browse Relation Attributes
FUNC	BTXE	TEXTBE	Browse Text for Entity
FUNC	BTXR	TEXTBR	Browse Text for Relation
FUNC	CO	HMFCOS0	Change Ownership
FUNC	DE	DELETEE	Delete an Entity
FUNC	DR	DELETER	Delete a Relation
FUNC	EW	HMFXTWU	Extended Where Used Report
FUNC	K	HMFKEY0	Define Keywords
FUNC	LE	LISTE	List Entity Instances
FUNC	LOCK	HMFUOLK	Lock an Entity
FUNC	LR	LISTR	List Relation Instances
FUNC	ME	MAINTE	Maintain Entity Attributes
FUNC	MR	MAINTR	Maintain Relation Attributes
FUNC	REFRESH	REFRESH	Refresh Entity
FUNC	REP	HMFREP	Produce Report
FUNC	RES	RESGLBL	View Results of Actions
FUNC	SA	SAVEAS	Save As Current Entity
FUNC	SRCH	SEARCH	Search for Keywords
FUNC	TXE	TEXTE	Define Text for Entity
FUNC	TXR	TEXTR	Define Text for Relation
FUNC	UEMTH	UEMTH	HPS User Entity Method
FUNC	UNLOCK	HMFUOFR	Unlock an Entity
FUNC	URMTH	URMTH	HPS User Relation Method
FVALUE	ACTIONS	ACTIONS	List Entity Actions
FVALUE	ADDE	ADDE	Add an Entity
FVALUE	ADDR	ADDR	Add a Relation
FVALUE	BE	BE	Browse Entity Attributes
FVALUE	BK	HMFKEYB	Browse Keywords
FVALUE	BR	BR	Browse Relation Attributes
FVALUE	BTXE	TEXTBE	Browse Text for Entity

FVALUE	BTXR	TEXTBR	Browse Text for Relation
FVALUE	CO	HMF COS0	Change Ownership
FVALUE	DE	DELETEE	Delete an Entity
FVALUE	DR	DELETER	Delete a Relation
FVALUE	EW	HMF XWU	Extended Where Used Report
FVALUE	K	HMF MKEY0	Define Keywords
FVALUE	LE	LISTE	List Entity Instances
FVALUE	LOCK	HMF UOLK	Lock an Entity
FVALUE	LR	LISTR	List Relation Instances
FVALUE	ME	MAINTE	Maintain Entity Attributes
FVALUE	MR	MAINTR	Maintain Relation Attributes
FVALUE	REFRESH	REFRESH	Refresh Entity
FVALUE	RES	RESGLBL	View Results of Actions
FVALUE	SA	SAVEAS	Save As Current Entity
FVALUE	SRCH	SEARCH	Search for Keywords
FVALUE	TXE	TEXTE	Define Text for Entity
FVALUE	TXR	TEXTR	Define Text for Relation
FVALUE	UEMTH	UEMTH	HPS User Entity Method
FVALUE	UNLOCK	HMF UOFR	Unlock an Entity
FVALUE	URMTH	URMTH	HPS User Relation Method
G			
GROUP	ACTIONS	ACTIONS	List Entity Actions
GROUP	ADDE	ADDE	Add an Entity
GROUP	ADDR	ADDR	Add a Relation
GROUP	BE	BE	Browse Entity Attributes
GROUP	BK	HMF MKEYB	Browse Keywords
GROUP	BR	BR	Browse Relation Attributes
GROUP	BTXE	TEXTBE	Browse Text for Entity
GROUP	BTXR	TEXTBR	Browse Text for Relation
GROUP	CO	HMF COS0	Change Ownership
GROUP	DE	DELETEE	Delete an Entity
GROUP	DR	DELETER	Delete a Relation
GROUP	EW	HMF XWU	Extended Where Used Report
GROUP	K	HMF MKEY0	Define Keywords
GROUP	LE	LISTE	List Entity Instances
GROUP	LOCK	HMF UOLK	Lock an Entity
GROUP	LR	LISTR	List Relation Instances
GROUP	ME	MAINTE	Maintain Entity Attributes
GROUP	MR	MAINTR	Maintain Relation Attributes

GROUP	REFRESH	REFRESH	Refresh Entity
GROUP	RES	RESGLBL	View Results of Actions
GROUP	SA	SAVEAS	Save As Current Entity
GROUP	SRCH	SEARCH	Search for Keywords
GROUP	TXE	TEXTE	Define Text for Entity
GROUP	TXR	TEXTR	Define Text for Relation
GROUP	UEMTH	UEMTH	HPS User Entity Method
GROUP	UNLOCK	HMFUOFR	Unlock an Entity
GROUP	URMTH	URMTH	HPS User Relation Method
H			
HELP	EW	HMFYWU	Extended Where Used Report
HLPTXT	ACTIONS	ACTIONS	List Entity Actions
HLPTXT	ADDE	ADDE	Add an Entity
HLPTXT	ADDR	ADDR	Add a Relation
HLPTXT	BE	BE	Browse Entity Attributes
HLPTXT	BK	HMFMYEYB	Browse Keywords
HLPTXT	BR	BR	Browse Relation Attributes
HLPTXT	BTXE	TEXTBE	Browse Text for Entity
HLPTXT	BTXR	TEXTBR	Browse Text for Relation
HLPTXT	CO	HMFYOS0	Change Ownership
HLPTXT	DE	DELETEE	Delete an Entity
HLPTXT	DR	DELETER	Delete a Relation
HLPTXT	EW	HMFYWU	Extended Where Used Report
HLPTXT	K	HMFMYEY0	Define Keywords
HLPTXT	LE	LISTE	List Entity Instances
HLPTXT	LOCK	HMFYOLK	Lock an Entity
HLPTXT	LR	LISTR	List Relation Instances
HLPTXT	ME	MAINTE	Maintain Entity Attributes
HLPTXT	MR	MAINTR	Maintain Relation Attributes
HLPTXT	REFRESH	REFRESH	Refresh Entity
HLPTXT	RES	RESGLBL	View Results of Actions
HLPTXT	SA	SAVEAS	Save As Current Entity
HLPTXT	SRCH	SEARCH	Search for Keywords
HLPTXT	TXE	TEXTE	Define Text for Entity
HLPTXT	TXR	TEXTR	Define Text for Relation
HLPTXT	UEMTH	UEMTH	HPS User Entity Method
HLPTXT	UNLOCK	HMFUOFR	Unlock an Entity
HLPTXT	URMTH	URMTH	HPS User Relation Method
I			

IDENT	ACTIONS	ACTIONS	List Entity Actions
IDENT	ADDE	ADDE	Add an Entity
IDENT	ADDR	ADDR	Add a Relation
IDENT	BE	BE	Browse Entity Attributes
IDENT	BK	HMFKEYB	Browse Keywords
IDENT	BR	BR	Browse Relation Attributes
IDENT	BTXE	TEXTBE	Browse Text for Entity
IDENT	BTXR	TEXTBR	Browse Text for Relation
IDENT	CO	HMFOS0	Change Ownership
IDENT	DE	DELETEE	Delete an Entity
IDENT	DR	DELETER	Delete a Relation
IDENT	EW	HMFYWU	Extended Where Used Report
IDENT	K	HMFKEY0	Define Keywords
IDENT	LE	LISTE	List Entity Instances
IDENT	LOCK	HMFUOLK	Lock an Entity
IDENT	LR	LISTR	List Relation Instances
IDENT	ME	MAINTE	Maintain Entity Attributes
IDENT	MR	MAINTR	Maintain Relation Attributes
IDENT	REFRESH	REFRESH	Refresh Entity
IDENT	RES	RESGLBL	View Results of Actions
IDENT	SA	SAVEAS	Save As Current Entity
IDENT	SRCH	SEARCH	Search for Keywords
IDENT	TXE	TEXTE	Define Text for Entity
IDENT	TXR	TEXTR	Define Text for Relation
IDENT	UEMTH	UEMTH	HPS User Entity Method
IDENT	UNLOCK	HMFUOFR	Unlock an Entity
IDENT	URMTH	URMTH	HPS User Relation Method
K			
KEY	ACTIONS	ACTIONS	List Entity Actions
KEY	ADDE	ADDE	Add an Entity
KEY	ADDR	ADDR	Add a Relation
KEY	BE	BE	Browse Entity Attributes
KEY	BK	HMFKEYB	Browse Keywords
KEY	BR	BR	Browse Relation Attributes
KEY	BTXE	TEXTBE	Browse Text for Entity
KEY	BTXR	TEXTBR	Browse Text for Relation
KEY	CO	HMFOS0	Change Ownership
KEY	DE	DELETEE	Delete an Entity
KEY	DR	DELETER	Delete a Relation

KEY	EW	HMFYWU	Extended Where Used Report
KEY	K	HMFMYEY0	Define Keywords
KEY	LE	LISTE	List Entity Instances
KEY	LOCK	HMFYOLK	Lock an Entity
KEY	LR	LISTR	List Relation Instances
KEY	ME	MAINTE	Maintain Entity Attributes
KEY	MR	MAINTR	Maintain Relation Attributes
KEY	REFRESH	REFRESH	Refresh Entity
KEY	RES	RESGLBL	View Results of Actions
KEY	SA	SAVEAS	Save As Current Entity
KEY	SRCH	SEARCH	Search for Keywords
KEY	TXE	TEXTE	Define Text for Entity
KEY	TXR	TEXTR	Define Text for Relation
KEY	UEMTH	UEMTH	HPS User Entity Method
KEY	UNLOCK	HMFYOFR	Unlock an Entity
KEY	URMTH	URMTH	HPS User Relation Method
L			
LANG	ACTIONS	ACTIONS	List Entity Actions
LANG	ADDE	ADDE	Add an Entity
LANG	ADDR	ADDR	Add a Relation
LANG	BE	BE	Browse Entity Attributes
LANG	BK	HMFMYEYB	Browse Keywords
LANG	BR	BR	Browse Relation Attributes
LANG	BTXE	TEXTBE	Browse Text for Entity
LANG	BTXR	TEXTBR	Browse Text for Relation
LANG	CO	HMFCOS0	Change Ownership
LANG	DE	DELETEE	Delete an Entity
LANG	DR	DELETER	Delete a Relation
LANG	K	HMFMYEY0	Define Keywords
LANG	LE	LISTE	List Entity Instances
LANG	LOCK	HMFYOLK	Lock an Entity
LANG	LR	LISTR	List Relation Instances
LANG	ME	MAINTE	Maintain Entity Attributes
LANG	MR	MAINTR	Maintain Relation Attributes
LANG	REFRESH	REFRESH	Refresh Entity
LANG	SA	SAVEAS	Save As Current Entity
LANG	SRCH	SEARCH	Search for Keywords
LANG	TXE	TEXTE	Define Text for Entity
LANG	TXR	TEXTR	Define Text for Relation

LANG	UEMTH	UEMTH	HPS User Entity Method
LANG	UNLOCK	HMFUOFR	Unlock an Entity
LANG	URMTH	URMTH	HPS User Relation Method
LOC	ACTIONS	ACTIONS	List Entity Actions
LOC	ADDE	ADDE	Add an Entity
LOC	ADDR	ADDR	Add a Relation
LOC	BE	BE	Browse Entity Attributes
LOC	BK	HMFMKEYB	Browse Keywords
LOC	BR	BR	Browse Relation Attributes
LOC	BTXE	TEXTBE	Browse Text for Entity
LOC	BTXR	TEXTBR	Browse Text for Relation
LOC	CO	HMFCOS0	Change Ownership
LOC	DE	DELETEE	Delete an Entity
LOC	DR	DELETER	Delete a Relation
LOC	EW	HMFYWU	Extended Where Used Report
LOC	K	HMFMKEY0	Define Keywords
LOC	LE	LISTE	List Entity Instances
LOC	LOCK	HMFUOLK	Lock an Entity
LOC	LR	LISTR	List Relation Instances
LOC	ME	MAINTE	Maintain Entity Attributes
LOC	MR	MAINTR	Maintain Relation Attributes
LOC	REFRESH	REFRESH	Refresh Entity
LOC	RES	RESGLBL	View Results of Actions
LOC	SA	SAVEAS	Save As Current Entity
LOC	SRCH	SEARCH	Search for Keywords
LOC	TXE	TEXTE	Define Text for Entity
LOC	TXR	TEXTR	Define Text for Relation
LOC	UEMTH	UEMTH	HPS User Entity Method
LOC	UNLOCK	HMFUOFR	Unlock an Entity
LOC	URMTH	URMTH	HPS User Relation Method
LOGPRO	ACTIONS	ACTIONS	List Entity Actions
LOGPRO	ADDE	ADDE	Add an Entity
LOGPRO	ADDR	ADDR	Add a Relation
LOGPRO	BE	BE	Browse Entity Attributes
LOGPRO	BK	HMFMKEYB	Browse Keywords
LOGPRO	BR	BR	Browse Relation Attributes
LOGPRO	BTXE	TEXTBE	Browse Text for Entity
LOGPRO	BTXR	TEXTBR	Browse Text for Relation
LOGPRO	CO	HMFCOS0	Change Ownership

LOGPRO	DE	DELETEE	Delete an Entity
LOGPRO	DR	DELETER	Delete a Relation
LOGPRO	EW	HMFXTWU	Extended Where Used Report
LOGPRO	K	HMFKEY0	Define Keywords
LOGPRO	LE	LISTE	List Entity Instances
LOGPRO	LOCK	HMFUOLK	Lock an Entity
LOGPRO	LR	LISTR	List Relation Instances
LOGPRO	ME	MAINTE	Maintain Entity Attributes
LOGPRO	MR	MAINTR	Maintain Relation Attributes
LOGPRO	REFRESH	REFRESH	Refresh Entity
LOGPRO	RES	RESGLBL	View Results of Actions
LOGPRO	SA	SAVEAS	Save As Current Entity
LOGPRO	SRCH	SEARCH	Search for Keywords
LOGPRO	TXE	TEXTE	Define Text for Entity
LOGPRO	TXR	TEXTR	Define Text for Relation
LOGPRO	UEMTH	UEMTH	HPS User Entity Method
LOGPRO	UNLOCK	HMFUOFR	Unlock an Entity
LOGPRO	URMTH	URMTH	HPS User Relation Method
M			
MACHINE	ACTIONS	ACTIONS	List Entity Actions
MACHINE	ADDE	ADDE	Add an Entity
MACHINE	ADDR	ADDR	Add a Relation
MACHINE	B	HFMEDB	Browse Source Code
MACHINE	BE	BE	Browse Entity Attributes
MACHINE	BK	HMFKEYB	Browse Keywords
MACHINE	BR	BR	Browse Relation Attributes
MACHINE	BTXE	TEXTBE	Browse Text for Entity
MACHINE	BTXR	TEXTBR	Browse Text for Relation
MACHINE	CO	HMFOS0	Change Ownership
MACHINE	DE	DELETEE	Delete an Entity
MACHINE	DISPCFG	DISPCFG	View Machine Configuration
MACHINE	DR	DELETER	Delete a Relation
MACHINE	EW	HMFXTWU	Extended Where Used Report
MACHINE	K	HMFKEY0	Define Keywords
MACHINE	LE	LISTE	List Entity Instances
MACHINE	LOCK	HMFUOLK	Lock an Entity
MACHINE	LR	LISTR	List Relation Instances
MACHINE	ME	MAINTE	Maintain Entity Attributes
MACHINE	MR	MAINTR	Maintain Relation Attributes

MACHINE	REFRESH	REFRESH	Refresh Entity
MACHINE	RES	RESGLBL	View Results of Actions
MACHINE	S	HMFMEDS	Define Source Code
MACHINE	SA	SAVEAS	Save As Current Entity
MACHINE	SRCH	SEARCH	Search for Keywords
MACHINE	TXE	TEXTE	Define Text for Entity
MACHINE	TXR	TEXTR	Define Text for Relation
MACHINE	UEMTH	UEMTH	HPS User Entity Method
MACHINE	UNLOCK	HMFUOFR	Unlock an Entity
MACHINE	URMTH	URMTH	HPS User Relation Method
MIGRTN	ACTIONS	ACTIONS	List Entity Actions
MIGRTN	ADDE	ADDE	Add an Entity
MIGRTN	ADDR	ADDR	Add a Relation
MIGRTN	ANALYZE	ANALYZE	Analyze Migration Impact
MIGRTN	APPROVE	SIGNOFF	Approve a Migration Process
MIGRTN	BE	BE	Browse Entity Attributes
MIGRTN	BK	HFMFMKEYB	Browse Keywords
MIGRTN	BR	BR	Browse Relation Attributes
MIGRTN	BTXE	TEXTBE	Browse Text for Entity
MIGRTN	BTXR	TEXTBR	Browse Text for Relation
MIGRTN	CLEANUP	CLEANUP	Clean-Up Migration Information
MIGRTN	CO	HMFCOS0	Change Ownership
MIGRTN	COPYAPP	COPYAPP	Copy Application Executables
MIGRTN	COPYMIG	COPYMIG	Copy Migration Files
MIGRTN	COUNT	COUNT	Summary Counts for Migration
MIGRTN	DE	DELETEE	Delete an Entity
MIGRTN	DR	DELETER	Delete a Relation
MIGRTN	EW	HMFXTWU	Extended Where Used Report
MIGRTN	EXPORT	EXPORT	Export Migration
MIGRTN	IMPORT	IMPORT	Import Migration
MIGRTN	K	HFMFMKEY0	Define Keywords
MIGRTN	LE	LISTE	List Entity Instances
MIGRTN	LISTMIG	LISTMIG	List Migration Contents
MIGRTN	LOCK	HMFUOLK	Lock an Entity
MIGRTN	LR	LISTR	List Relation Instances
MIGRTN	ME	MAINTE	Maintain Entity Attributes
MIGRTN	MIGRATE	MIGRATE	Automated Migration Facility
MIGRTN	MR	MAINTR	Maintain Relation Attributes
MIGRTN	RD	GARDX00	Analyze Rebuild Impact

MIGRTN	REFRESH	REFRESH	Refresh Entity
MIGRTN	REP	REP	Report on a Migration
MIGRTN	RES	RESMIG	View Results of Mig Actions
MIGRTN	SRCH	SEARCH	Search for Keywords
MIGRTN	TXE	TEXTE	Define Text for Entity
MIGRTN	TXR	TEXTR	Define Text for Relation
MIGRTN	UEMTH	UEMTH	HPS User Entity Method
MIGRTN	UNLOCK	HMFUOFR	Unlock an Entity
MIGRTN	URMTH	URMTH	HPS User Relation Method
O			
ORG	ACTIONS	ACTIONS	List Entity Actions
ORG	ADDE	ADDE	Add an Entity
ORG	ADDR	ADDR	Add a Relation
ORG	BE	BE	Browse Entity Attributes
ORG	BK	HMFKEYB	Browse Keywords
ORG	BR	BR	Browse Relation Attributes
ORG	BTXE	TEXTBE	Browse Text for Entity
ORG	BTXR	TEXTBR	Browse Text for Relation
ORG	CO	HMFCOS0	Change Ownership
ORG	DE	DELETEE	Delete an Entity
ORG	DR	DELETER	Delete a Relation
ORG	EW	HMFYWU	Extended Where Used Report
ORG	K	HMFKEY0	Define Keywords
ORG	LE	LISTE	List Entity Instances
ORG	LOCK	HMFUOLK	Lock an Entity
ORG	LR	LISTR	List Relation Instances
ORG	ME	MAINTE	Maintain Entity Attributes
ORG	MR	MAINTR	Maintain Relation Attributes
ORG	REFRESH	REFRESH	Refresh Entity
ORG	RES	RESGLBL	View Results of Actions
ORG	SA	SAVEAS	Save As Current Entity
ORG	SRCH	SEARCH	Search for Keywords
ORG	TXE	TEXTE	Define Text for Entity
ORG	TXR	TEXTR	Define Text for Relation
ORG	UEMTH	UEMTH	HPS User Entity Method
ORG	UNLOCK	HMFUOFR	Unlock an Entity
ORG	URMTH	URMTH	HPS User Relation Method
P			
PARTITION	ACTIONS	ACTIONS	List Entity Actions

PARTITION	ACTIVATE	ACTIVATE	Activate Configuration Unit
PARTITION	ADDE	ADDE	Add an Entity
PARTITION	ADDR	ADDR	Add a Relation
PARTITION	BE	BE	Browse Entity Attributes
PARTITION	BK	HMFKEYB	Browse Keywords
PARTITION	BR	BR	Browse Relation Attributes
PARTITION	BTXE	TEXTBE	Browse Text for Entity
PARTITION	BTXR	TEXTBR	Browse Text for Relation
PARTITION	CO	HMFOS0	Change Ownership
PARTITION	DE	DELETEE	Delete an Entity
PARTITION	DR	DELETER	Delete a Relation
PARTITION	EW	HMFOWU	Extended Where Used Report
PARTITION	K	HMFKEY0	Define Keywords
PARTITION	LE	LISTE	List Entity Instances
PARTITION	LOCK	HMFOLK	Lock an Entity
PARTITION	LR	LISTR	List Relation Instances
PARTITION	ME	MAINTE	Maintain Entity Attributes
PARTITION	MR	MAINTR	Maintain Relation Attributes
PARTITION	REFRESH	REFRESH	Refresh Entity
PARTITION	RES	RESGLBL	View Results of Actions
PARTITION	SA	SAVEAS	Save As Current Entity
PARTITION	SRCH	SEARCH	Search for Keywords
PARTITION	TXE	TEXTE	Define Text for Entity
PARTITION	TXR	TEXTR	Define Text for Relation
PARTITION	UEMTH	UEMTH	HPS User Entity Method
PARTITION	UNLOCK	HMFUOFR	Unlock an Entity
PARTITION	URMTH	URMTH	HPS User Relation Method
PHYEVT	ACTIONS	ACTIONS	List Entity Actions
PHYEVT	ADDE	ADDE	Add an Entity
PHYEVT	ADDR	ADDR	Add a Relation
PHYEVT	BE	BE	Browse Entity Attributes
PHYEVT	BK	HMFKEYB	Browse Keywords
PHYEVT	BR	BR	Browse Relation Attributes
PHYEVT	BTXE	TEXTBE	Browse Text for Entity
PHYEVT	BTXR	TEXTBR	Browse Text for Relation
PHYEVT	CO	HMFOS0	Change Ownership
PHYEVT	DE	DELETEE	Delete an Entity
PHYEVT	DR	DELETER	Delete a Relation
PHYEVT	EW	HMFOWU	Extended Where Used Report

PHYEVT	K	HMFKEY0	Define Keywords
PHYEVT	LE	LISTE	List Entity Instances
PHYEVT	LOCK	HMFUOLK	Lock an Entity
PHYEVT	LR	LISTR	List Relation Instances
PHYEVT	ME	MAINTE	Maintain Entity Attributes
PHYEVT	MR	MAINTR	Maintain Relation Attributes
PHYEVT	REFRESH	REFRESH	Refresh Entity
PHYEVT	RES	RESGLBL	View Results of Actions
PHYEVT	SA	SAVEAS	Save As Current Entity
PHYEVT	SRCH	SEARCH	Search for Keywords
PHYEVT	TXE	TEXTE	Define Text for Entity
PHYEVT	TXR	TEXTR	Define Text for Relation
PHYEVT	UEMTH	UEMTH	HPS User Entity Method
PHYEVT	UNLOCK	HMFUOFR	Unlock an Entity
PHYEVT	URMTH	URMTH	HPS User Relation Method
PROCES	ACTIONS	ACTIONS	List Entity Actions
PROCES	ADDE	ADDE	Add an Entity
PROCES	ADDR	ADDR	Add a Relation
PROCES	ASSIGNCU	HMFACU0	Assign Configuration Unit
PROCES	BE	BE	Browse Entity Attributes
PROCES	BK	HMFKEYB	Browse Keywords
PROCES	BR	BR	Browse Relation Attributes
PROCES	BTXE	TEXTBE	Browse Text for Entity
PROCES	BTXR	TEXTBR	Browse Text for Relation
PROCES	CO	HMFCOS0	Change Ownership
PROCES	DE	DELETEE	Delete an Entity
PROCES	DR	DELETER	Delete a Relation
PROCES	EW	HMFYWU	Extended Where Used Report
PROCES	K	HMFKEY0	Define Keywords
PROCES	LE	LISTE	List Entity Instances
PROCES	LISTRBD	LISTRBD	List Rebuild Contents
PROCES	LOCK	HMFUOLK	Lock an Entity
PROCES	LR	LISTR	List Relation Instances
PROCES	ME	MAINTE	Maintain Entity Attributes
PROCES	MR	MAINTR	Maintain Relation Attributes
PROCES	PR	PREPPROC	Prepare Execution Environment
PROCES	PRB	PREPPRCB	Prepare Exec Env (background)
PROCES	REFRESH	REFRESH	Refresh Entity
PROCES	REP	HMFREP	Produce Report

PROCES	RES	RESGLBL	View Results of Actions
PROCES	SA	SAVEAS	Save As Current Entity
PROCES	SRCH	SEARCH	Search for Keywords
PROCES	SUPERPR	SUPERPR	Super Prepare
PROCES	TXE	TEXTE	Define Text for Entity
PROCES	TXR	TEXTR	Define Text for Relation
PROCES	UEMTH	UEMTH	HPS User Entity Method
PROCES	UNLOCK	HMFUOFR	Unlock an Entity
PROCES	URMTH	URMTH	HPS User Relation Method
PROJCT	ACTIONS	ACTIONS	List Entity Actions
PROJCT	ADDE	ADDE	Add an Entity
PROJCT	ADDR	ADDR	Add a Relation
PROJCT	BE	BE	Browse Entity Attributes
PROJCT	BK	HFMKEYB	Browse Keywords
PROJCT	BR	BR	Browse Relation Attributes
PROJCT	BTXE	TEXTBE	Browse Text for Entity
PROJCT	BTXR	TEXTBR	Browse Text for Relation
PROJCT	DE	DELETEE	Delete an Entity
PROJCT	DR	DELETER	Delete a Relation
PROJCT	EW	HMFYWU	Extended Where Used Report
PROJCT	IMS	IMSUTIL	IMS Utilities
PROJCT	K	HFMKEY0	Define Keywords
PROJCT	LE	LISTE	List Entity Instances
PROJCT	LOCK	HMFUOLK	Lock an Entity
PROJCT	LR	LISTR	List Relation Instances
PROJCT	ME	MAINTE	Maintain Entity Attributes
PROJCT	MR	MAINTR	Maintain Relation Attributes
PROJCT	REFRESH	REFRESH	Refresh Entity
PROJCT	RES	RESGLBL	View Results of Actions
PROJCT	SA	SAVEAS	Save As Current Entity
PROJCT	SRCH	SEARCH	Search for Keywords
PROJCT	TXE	TEXTE	Define Text for Entity
PROJCT	TXR	TEXTR	Define Text for Relation
PROJCT	UEMTH	UEMTH	HPS User Entity Method
PROJCT	UNLOCK	HMFUOFR	Unlock an Entity
PROJCT	URMTH	URMTH	HPS User Relation Method
R			
RBDPKG	ACTIONS	ACTIONS	List Entity Actions
RBDPKG	ADDE	ADDE	Add an Entity

RBDPKG	ADDR	ADDR	Add a Relation
RBDPKG	BE	BE	Browse Entity Attributes
RBDPKG	BK	HMFKEYB	Browse Keywords
RBDPKG	BR	BR	Browse Relation Attributes
RBDPKG	BTXE	TEXTBE	Browse Text for Entity
RBDPKG	CLEANUP	RBDCLUP	Clean-Up Rebuild Information
RBDPKG	CO	HMFOS0	Change Ownership
RBDPKG	DE	DELETEE	Delete an Entity
RBDPKG	DR	DELETER	Delete a Relation
RBDPKG	EW	HMFYWU	Extended Where Used Report
RBDPKG	EXPORT	EXPORT	Export Migration
RBDPKG	K	HMFKEY0	Define Keywords
RBDPKG	LE	LISTE	List Entity Instances
RBDPKG	LISTMIG	LISTMIG	List Migration Contents
RBDPKG	LISTRBD	LISTRBD	List Rebuild Contents
RBDPKG	LOCK	HMFUOLK	Lock an Entity
RBDPKG	LR	LISTR	List Relation Instances
RBDPKG	ME	MAINTE	Maintain Entity Attributes
RBDPKG	MR	MAINTR	Maintain Relation Attributes
RBDPKG	RANALYZE	RANALYZE	Rebuild Analyze
RBDPKG	REFRESH	REFRESH	Refresh Entity
RBDPKG	REP	RREPORT	Report
RBDPKG	RES	RESRBD	View Results of Rebuild
RBDPKG	RESETINS	RESETINS	Reset Install
RBDPKG	RESETPR	RESETPR	Reset Prepare
RBDPKG	RINSTALL	RINSTALL	Rebuild Install
RBDPKG	RPREPARE	RPREPARE	Rebuild Prepare
RBDPKG	SA	SAVEAS	Save As Current Entity
RBDPKG	SRCH	SEARCH	Search for Keywords
RBDPKG	SUPERPR	SUPERPR	Super Prepare
RBDPKG	TXE	TEXTE	Define Text for Entity
RBDPKG	UEMTH	UEMTH	HPS User Entity Method
RBDPKG	UNLOCK	HMFUOFR	Unlock an Entity
RBDPKG	URMTH	URMTH	HPS User Relation Method
REPORT	ACTIONS	ACTIONS	List Entity Actions
REPORT	ADDE	ADDE	Add an Entity
REPORT	ADDR	ADDR	Add a Relation
REPORT	ASSIGNCU	HMFACU0	Assign Configuration Unit
REPORT	B	HFMEDB	Browse Source Code

REPORT	BE	BE	Browse Entity Attributes
REPORT	BK	HMFKEYB	Browse Keywords
REPORT	BR	BR	Browse Relation Attributes
REPORT	BTXE	TEXTBE	Browse Text for Entity
REPORT	BTXR	TEXTBR	Browse Text for Relation
REPORT	CO	HMFOS0	Change Ownership
REPORT	DE	DELETEE	Delete an Entity
REPORT	DR	DELETER	Delete a Relation
REPORT	EW	HMFWU	Extended Where Used Report
REPORT	K	HMFKEY0	Define Keywords
REPORT	LE	LISTE	List Entity Instances
REPORT	LISTRBD	LISTRBD	List Rebuild Contents
REPORT	LOCK	HMFOLK	Lock an Entity
REPORT	LR	LISTR	List Relation Instances
REPORT	ME	MAINTE	Maintain Entity Attributes
REPORT	MR	MAINTR	Maintain Relation Attributes
REPORT	PR	PREPREP	Prepare Report
REPORT	REFRESH	REFRESH	Refresh Entity
REPORT	REP	HMFREP	Produce Report
REPORT	RES	RESREP	View Results of Report Actions
REPORT	S	HFMEDS	Define Source Code
REPORT	SA	SAVEAS	Save As Current Entity
REPORT	SRCH	SEARCH	Search for Keywords
REPORT	SUPERPR	SUPERPR	Super Prepare
REPORT	TXE	TEXTE	Define Text for Entity
REPORT	TXR	TEXTR	Define Text for Relation
REPORT	UEMTH	UEMTH	HPS User Entity Method
REPORT	UNLOCK	HMFUFR	Unlock an Entity
REPORT	URMTH	URMTH	HPS User Relation Method
REPOSIT	ACTIONS	ACTIONS	List Entity Actions
REPOSIT	ADDE	ADDE	Add an Entity
REPOSIT	ADDR	ADDR	Add a Relation
REPOSIT	BE	BE	Browse Entity Attributes
REPOSIT	BK	HMFKEYB	Browse Keywords
REPOSIT	BR	BR	Browse Relation Attributes
REPOSIT	BTXE	TEXTBE	Browse Text for Entity
REPOSIT	BTXR	TEXTBR	Browse Text for Relation
REPOSIT	DE	DELETEE	Delete an Entity
REPOSIT	DR	DELETER	Delete a Relation

REPOSIT	EW	HMFXTWU	Extended Where Used Report
REPOSIT	K	HMFMTKEY0	Define Keywords
REPOSIT	LE	LISTE	List Entity Instances
REPOSIT	LOCK	HMFUOLK	Lock an Entity
REPOSIT	LR	LISTR	List Relation Instances
REPOSIT	ME	MAINTE	Maintain Entity Attributes
REPOSIT	MR	MAINTR	Maintain Relation Attributes
REPOSIT	REFRESH	REFRESH	Refresh Entity
REPOSIT	RES	RESGLBL	View Results of Actions
REPOSIT	SA	SAVEAS	Save As Current Entity
REPOSIT	SRCH	SEARCH	Search for Keywords
REPOSIT	TXE	TEXTE	Define Text for Entity
REPOSIT	TXR	TEXTR	Define Text for Relation
REPOSIT	UEMTH	UEMTH	HPS User Entity Method
REPOSIT	UNLOCK	HMFUOFR	Unlock an Entity
REPOSIT	URMTH	URMTH	HPS User Relation Method
RLN	ACTIONS	ACTIONS	List Entity Actions
RLN	ADDE	ADDE	Add an Entity
RLN	ADDR	ADDR	Add a Relation
RLN	BE	BE	Browse Entity Attributes
RLN	BK	HMFMTKEYB	Browse Keywords
RLN	BR	BR	Browse Relation Attributes
RLN	BTXE	TEXTBE	Browse Text for Entity
RLN	BTXR	TEXTBR	Browse Text for Relation
RLN	CO	HMFCOS0	Change Ownership
RLN	DE	DELETEE	Delete an Entity
RLN	DR	DELETER	Delete a Relation
RLN	EW	HMFXTWU	Extended Where Used Report
RLN	K	HMFMTKEY0	Define Keywords
RLN	LE	LISTE	List Entity Instances
RLN	LOCK	HMFUOLK	Lock an Entity
RLN	LR	LISTR	List Relation Instances
RLN	ME	MAINTE	Maintain Entity Attributes
RLN	MR	MAINTR	Maintain Relation Attributes
RLN	REFRESH	REFRESH	Refresh Entity
RLN	RES	RESGLBL	View Results of Actions
RLN	SA	SAVEAS	Save As Current Entity
RLN	SRCH	SEARCH	Search for Keywords
RLN	TXE	TEXTE	Define Text for Entity

RLN	TXR	TEXTR	Define Text for Relation
RLN	UEMTH	UEMTH	HPS User Entity Method
RLN	UNLOCK	HMFUOFR	Unlock an Entity
RLN	URMTH	URMTH	HPS User Relation Method
RULE	ACTIONS	ACTIONS	List Entity Actions
RULE	ADDE	ADDE	Add an Entity
RULE	ADDR	ADDR	Add a Relation
RULE	ASSIGNCU	HMFACU0	Assign Configuration Unit
RULE	B	HFMEDB	Browse Source Code
RULE	BE	BE	Browse Entity Attributes
RULE	BINDPKG	BINDPKG	Bind DB2 Package
RULE	BK	HFMKEYB	Browse Keywords
RULE	BR	BR	Browse Relation Attributes
RULE	BTS	BTS	Batch Terminal Simulation
RULE	BTXE	TEXTBE	Browse Text for Entity
RULE	BTXR	TEXTBR	Browse Text for Relation
RULE	CO	HMFCOS0	Change Ownership
RULE	DE	DELETEE	Delete an Entity
RULE	DR	DELETER	Delete a Relation
RULE	DYNAMIC	HMFDYLK	Establish Dynamic Linkage
RULE	EW	HMFYWU	Extended Where Used Report
RULE	K	HFMKEY0	Define Keywords
RULE	LE	LISTE	List Entity Instances
RULE	LISTRBD	LISTRBD	List Rebuild Contents
RULE	LOCK	HMFUOLK	Lock an Entity
RULE	LR	LISTR	List Relation Instances
RULE	ME	MAINTE	Maintain Entity Attributes
RULE	MR	MAINTR	Maintain Relation Attributes
RULE	PR	PREPRULE	Prepare Rule
RULE	PSB	MODPSB	Modify PSB (IMS)
RULE	RBD	RBDRULE	Rebuild Entity
RULE	RDTL	RDTL	IMS Rule Processing Detail
RULE	REFRESH	REFRESH	Refresh Entity
RULE	REP	HMFREP	Produce Report
RULE	RES	RESRULE	View Results of Rule Actions
RULE	S	HFMEDS	Define Source Code
RULE	SA	SAVEAS	Save As Current Entity
RULE	SRCH	SEARCH	Search for Keywords
RULE	STATIC	HMFSTLK	Establish Static Linkage

RULE	SUPERPR	SUPERPR	Super Prepare
RULE	TE	HMFMBTST	Test Batch Entity
RULE	TXE	TEXTE	Define Text for Entity
RULE	TXR	TEXTR	Define Text for Relation
RULE	UEMTH	UEMTH	HPS User Entity Method
RULE	UNLOCK	HMFUOFR	Unlock an Entity
RULE	URMTH	URMTH	HPS User Relation Method
RULE	VER	HMFLEXO	Verify Hierarchy
S			
SECTN	ACTIONS	ACTIONS	List Entity Actions
SECTN	ADDE	ADDE	Add an Entity
SECTN	ADDR	ADDR	Add a Relation
SECTN	B	HFMEDB	Browse Source Code
SECTN	BE	BE	Browse Entity Attributes
SECTN	BK	HFMKEYB	Browse Keywords
SECTN	BR	BR	Browse Relation Attributes
SECTN	BTXE	TEXTBE	Browse Text for Entity
SECTN	BTXR	TEXTBR	Browse Text for Relation
SECTN	CO	HMFCOS0	Change Ownership
SECTN	DE	DELETEE	Delete an Entity
SECTN	DR	DELETER	Delete a Relation
SECTN	EW	HMFYWU	Extended Where Used Report
SECTN	K	HFMKEY0	Define Keywords
SECTN	LE	LISTE	List Entity Instances
SECTN	LOCK	HMFUOLK	Lock an Entity
SECTN	LR	LISTR	List Relation Instances
SECTN	ME	MAINTE	Maintain Entity Attributes
SECTN	MR	MAINTR	Maintain Relation Attributes
SECTN	REFRESH	REFRESH	Refresh Entity
SECTN	REP	HMFREP	Produce Report
SECTN	RES	RESGLBL	View Results of Actions
SECTN	S	HFMEDS	Define Source Code
SECTN	SA	SAVEAS	Save As Current Entity
SECTN	SRCH	SEARCH	Search for Keywords
SECTN	TXE	TEXTE	Define Text for Entity
SECTN	TXR	TEXTR	Define Text for Relation
SECTN	UEMTH	UEMTH	HPS User Entity Method
SECTN	UNLOCK	HMFUOFR	Unlock an Entity
SECTN	URMTH	URMTH	HPS User Relation Method

SERVER	ACTIONS	ACTIONS	List Entity Actions
SERVER	ADDE	ADDE	Add an Entity
SERVER	ADDR	ADDR	Add a Relation
SERVER	B	HMFMEDB	Browse Source Code
SERVER	BE	BE	Browse Entity Attributes
SERVER	BK	HFMKEYB	Browse Keywords
SERVER	BR	BR	Browse Relation Attributes
SERVER	BTXE	TEXTBE	Browse Text for Entity
SERVER	BTXR	TEXTBR	Browse Text for Relation
SERVER	CO	HMFCOS0	Change Ownership
SERVER	DE	DELETEE	Delete an Entity
SERVER	DR	DELETER	Delete a Relation
SERVER	EW	HMFYWU	Extended Where Used Report
SERVER	K	HFMKEY0	Define Keywords
SERVER	LE	LISTE	List Entity Instances
SERVER	LOCK	HMFUOLK	Lock an Entity
SERVER	LR	LISTR	List Relation Instances
SERVER	ME	MAINTE	Maintain Entity Attributes
SERVER	MR	MAINTR	Maintain Relation Attributes
SERVER	REFRESH	REFRESH	Refresh Entity
SERVER	RES	RESGLBL	View Results of Actions
SERVER	S	HFMEDS	Define Source Code
SERVER	SA	SAVEAS	Save As Current Entity
SERVER	SRCH	SEARCH	Search for Keywords
SERVER	TXE	TEXTE	Define Text for Entity
SERVER	TXR	TEXTR	Define Text for Relation
SERVER	UEMTH	UEMTH	HPS User Entity Method
SERVER	UNLOCK	HMFUOFR	Unlock an Entity
SERVER	URMTH	URMTH	HPS User Relation Method
SET	ACTIONS	ACTIONS	List Entity Actions
SET	ADDE	ADDE	Add an Entity
SET	ADDR	ADDR	Add a Relation
SET	ASSIGNCU	HMFACU0	Assign Configuration Unit
SET	BE	BE	Browse Entity Attributes
SET	BK	HFMKEYB	Browse Keywords
SET	BR	BR	Browse Relation Attributes
SET	BTXE	TEXTBE	Browse Text for Entity
SET	BTXR	TEXTBR	Browse Text for Relation
SET	CO	HMFCOS0	Change Ownership

SET	DE	DELETEE	Delete an Entity
SET	DR	DELETER	Delete a Relation
SET	EW	HMFXTWU	Extended Where Used Report
SET	K	HMFKEY0	Define Keywords
SET	LE	LISTE	List Entity Instances
SET	LISTRBD	LISTRBD	List Rebuild Contents
SET	LOCK	HMFUOLK	Lock an Entity
SET	LR	LISTR	List Relation Instances
SET	ME	MAINTE	Maintain Entity Attributes
SET	MR	MAINTR	Maintain Relation Attributes
SET	PR	PREPSET	Prepare Set
SET	REFRESH	REFRESH	Refresh Entity
SET	REP	HMFREP	Produce Report
SET	RES	RESSET	View Results of Set Actions
SET	SA	SAVEAS	Save As Current Entity
SET	SRCH	SEARCH	Search for Keywords
SET	TXE	TEXTE	Define Text for Entity
SET	TXR	TEXTR	Define Text for Relation
SET	UEMTH	UEMTH	HPS User Entity Method
SET	UNLOCK	HMFUOFR	Unlock an Entity
SET	URMTH	URMTH	HPS User Relation Method
STATE	ACTIONS	ACTIONS	List Entity Actions
STATE	ADDE	ADDE	Add an Entity
STATE	ADDR	ADDR	Add a Relation
STATE	BE	BE	Browse Entity Attributes
STATE	BK	HMFKEYB	Browse Keywords
STATE	BR	BR	Browse Relation Attributes
STATE	BTXE	TEXTBE	Browse Text for Entity
STATE	BTXR	TEXTBR	Browse Text for Relation
STATE	CO	HMFCOS0	Change Ownership
STATE	DE	DELETEE	Delete an Entity
STATE	DR	DELETER	Delete a Relation
STATE	EW	HMFXTWU	Extended Where Used Report
STATE	K	HMFKEY0	Define Keywords
STATE	LE	LISTE	List Entity Instances
STATE	LOCK	HMFUOLK	Lock an Entity
STATE	LR	LISTR	List Relation Instances
STATE	ME	MAINTE	Maintain Entity Attributes
STATE	MR	MAINTR	Maintain Relation Attributes

STATE	REFRESH	REFRESH	Refresh Entity
STATE	RES	RESGLBL	View Results of Actions
STATE	SA	SAVEAS	Save As Current Entity
STATE	SRCH	SEARCH	Search for Keywords
STATE	TXE	TEXTE	Define Text for Entity
STATE	TXR	TEXTR	Define Text for Relation
STATE	UEMTH	UEMTH	HPS User Entity Method
STATE	UNLOCK	HMFUOFR	Unlock an Entity
STATE	URMTH	URMTH	HPS User Relation Method
STORE	ACTIONS	ACTIONS	List Entity Actions
STORE	ADDE	ADDE	Add an Entity
STORE	ADDR	ADDR	Add a Relation
STORE	BE	BE	Browse Entity Attributes
STORE	BK	HFMKEYB	Browse Keywords
STORE	BR	BR	Browse Relation Attributes
STORE	BTXE	TEXTBE	Browse Text for Entity
STORE	BTXR	TEXTBR	Browse Text for Relation
STORE	CO	HMFOS0	Change Ownership
STORE	DE	DELETEE	Delete an Entity
STORE	DR	DELETER	Delete a Relation
STORE	EW	HMFVWU	Extended Where Used Report
STORE	K	HFMKEY0	Define Keywords
STORE	LE	LISTE	List Entity Instances
STORE	LOCK	HMFUOLK	Lock an Entity
STORE	LR	LISTR	List Relation Instances
STORE	ME	MAINTE	Maintain Entity Attributes
STORE	MR	MAINTR	Maintain Relation Attributes
STORE	REFRESH	REFRESH	Refresh Entity
STORE	RES	RESGLBL	View Results of Actions
STORE	SA	SAVEAS	Save As Current Entity
STORE	SRCH	SEARCH	Search for Keywords
STORE	TXE	TEXTE	Define Text for Entity
STORE	TXR	TEXTR	Define Text for Relation
STORE	UEMTH	UEMTH	HPS User Entity Method
STORE	UNLOCK	HMFUOFR	Unlock an Entity
STORE	URMTH	URMTH	HPS User Relation Method
SYMBOL	ACTIONS	ACTIONS	List Entity Actions
SYMBOL	ADDE	ADDE	Add an Entity
SYMBOL	ADDR	ADDR	Add a Relation

SYMBOL	BE	BE	Browse Entity Attributes
SYMBOL	BK	HMFKEYB	Browse Keywords
SYMBOL	BR	BR	Browse Relation Attributes
SYMBOL	BTXE	TEXTBE	Browse Text for Entity
SYMBOL	BTXR	TEXTBR	Browse Text for Relation
SYMBOL	CO	HMFOS0	Change Ownership
SYMBOL	DE	DELETEE	Delete an Entity
SYMBOL	DR	DELETER	Delete a Relation
SYMBOL	EW	HMFOWU	Extended Where Used Report
SYMBOL	K	HMFKEY0	Define Keywords
SYMBOL	LE	LISTE	List Entity Instances
SYMBOL	LOCK	HMFOLK	Lock an Entity
SYMBOL	LR	LISTR	List Relation Instances
SYMBOL	ME	MAINTE	Maintain Entity Attributes
SYMBOL	MR	MAINTR	Maintain Relation Attributes
SYMBOL	REFRESH	REFRESH	Refresh Entity
SYMBOL	RES	RESGLBL	View Results of Actions
SYMBOL	SA	SAVEAS	Save As Current Entity
SYMBOL	SRCH	SEARCH	Search for Keywords
SYMBOL	TXE	TEXTE	Define Text for Entity
SYMBOL	TXR	TEXTR	Define Text for Relation
SYMBOL	UEMTH	UEMTH	HPS User Entity Method
SYMBOL	UNLOCK	HMFUFR	Unlock an Entity
SYMBOL	URMTH	URMTH	HPS User Relation Method
T			
TABLE	ACTIONS	ACTIONS	List Entity Actions
TABLE	ADDE	ADDE	Add an Entity
TABLE	ADDR	ADDR	Add a Relation
TABLE	BE	BE	Browse Entity Attributes
TABLE	BK	HMFKEYB	Browse Keywords
TABLE	BR	BR	Browse Relation Attributes
TABLE	BTXE	TEXTBE	Browse Text for Entity
TABLE	BTXR	TEXTBR	Browse Text for Relation
TABLE	CO	HMFOS0	Change Ownership
TABLE	DE	DELETEE	Delete an Entity
TABLE	DR	DELETER	Delete a Relation
TABLE	EW	HMFOWU	Extended Where Used Report
TABLE	K	HMFKEY0	Define Keywords
TABLE	LE	LISTE	List Entity Instances

TABLE	LOCK	HMFUOLK	Lock an Entity
TABLE	LR	LISTR	List Relation Instances
TABLE	ME	MAINTE	Maintain Entity Attributes
TABLE	MR	MAINTR	Maintain Relation Attributes
TABLE	REFRESH	REFRESH	Refresh Entity
TABLE	RES	RESGLBL	View Results of Actions
TABLE	SA	SAVEAS	Save As Current Entity
TABLE	SRCH	SEARCH	Search for Keywords
TABLE	TXE	TEXTE	Define Text for Entity
TABLE	TXR	TEXTR	Define Text for Relation
TABLE	UEMTH	UEMTH	HPS User Entity Method
TABLE	UNLOCK	HMFUOFR	Unlock an Entity
TABLE	URMTH	URMTH	HPS User Relation Method
TRANS	ACTIONS	ACTIONS	List Entity Actions
TRANS	ADDE	ADDE	Add an Entity
TRANS	ADDR	ADDR	Add a Relation
TRANS	BE	BE	Browse Entity Attributes
TRANS	BK	HMFKEYB	Browse Keywords
TRANS	BR	BR	Browse Relation Attributes
TRANS	BTXE	TEXTBE	Browse Text for Entity
TRANS	BTXR	TEXTBR	Browse Text for Relation
TRANS	CO	HMFCOS0	Change Ownership
TRANS	DE	DELETEE	Delete an Entity
TRANS	DR	DELETER	Delete a Relation
TRANS	EW	HMFXTWU	Extended Where Used Report
TRANS	K	HMFKEY0	Define Keywords
TRANS	LE	LISTE	List Entity Instances
TRANS	LOCK	HMFUOLK	Lock an Entity
TRANS	LR	LISTR	List Relation Instances
TRANS	ME	MAINTE	Maintain Entity Attributes
TRANS	MR	MAINTR	Maintain Relation Attributes
TRANS	REFRESH	REFRESH	Refresh Entity
TRANS	RES	RESGLBL	View Results of Actions
TRANS	SA	SAVEAS	Save As Current Entity
TRANS	SRCH	SEARCH	Search for Keywords
TRANS	TXE	TEXTE	Define Text for Entity
TRANS	TXR	TEXTR	Define Text for Relation
TRANS	UEMTH	UEMTH	HPS User Entity Method
TRANS	UNLOCK	HMFUOFR	Unlock an Entity

TRANS	URMTH	URMTH	HPS User Relation Method
U			
USER	ACTIONS	ACTIONS	List Entity Actions
USER	ADDE	ADDE	Add an Entity
USER	ADDR	ADDR	Add a Relation
USER	BE	BE	Browse Entity Attributes
USER	BK	HFMKEYB	Browse Keywords
USER	BR	BR	Browse Relation Attributes
USER	BTXE	TEXTBE	Browse Text for Entity
USER	BTXR	TEXTBR	Browse Text for Relation
USER	CO	HMFOS0	Change Ownership
USER	DE	DELETEE	Delete an Entity
USER	DR	DELETER	Delete a Relation
USER	EW	HMFOWU	Extended Where Used Report
USER	K	HFMKEY0	Define Keywords
USER	LE	LISTE	List Entity Instances
USER	LOCK	HMFOLK	Lock an Entity
USER	LR	LISTR	List Relation Instances
USER	ME	MAINTE	Maintain Entity Attributes
USER	MR	MAINTR	Maintain Relation Attributes
USER	REFRESH	REFRESH	Refresh Entity
USER	RES	RESGLBL	View Results of Actions
USER	SA	SAVEAS	Save As Current Entity
USER	SRCH	SEARCH	Search for Keywords
USER	TXE	TEXTE	Define Text for Entity
USER	TXR	TEXTR	Define Text for Relation
USER	UEMTH	UEMTH	User Entity Method
USER	UNLOCK	HMFUOFR	Unlock an Entity
USER	URMTH	URMTH	User Relation Method
V			
VALUES	ACTIONS	ACTIONS	List Entity Actions
VALUES	ADDE	ADDE	Add an Entity
VALUES	ADDR	ADDR	Add a Relation
VALUES	BE	BE	Browse Entity Attributes
VALUES	BK	HFMKEYB	Browse Keywords
VALUES	BR	BR	Browse Relation Attributes
VALUES	BTXE	TEXTBE	Browse Text for Entity
VALUES	BTXR	TEXTBR	Browse Text for Relation
VALUES	CO	HMFOS0	Change Ownership

VALUES	DE	DELETEE	Delete an Entity
VALUES	DR	DELETER	Delete a Relation
VALUES	EW	HMFXTWU	Extended Where Used Report
VALUES	K	HMFKEY0	Define Keywords
VALUES	LE	LISTE	List Entity Instances
VALUES	LOCK	HMFUOLK	Lock an Entity
VALUES	LR	LISTR	List Relation Instances
VALUES	ME	MAINTE	Maintain Entity Attributes
VALUES	MR	MAINTR	Maintain Relation Attributes
VALUES	REFRESH	REFRESH	Refresh Entity
VALUES	RES	RESGLBL	View Results of Actions
VALUES	SA	SAVEAS	Save As Current Entity
VALUES	SRCH	SEARCH	Search for Keywords
VALUES	TXE	TEXTE	Define Text for Entity
VALUES	TXR	TEXTR	Define Text for Relation
VALUES	UEMTH	UEMTH	User Entity Method
VALUES	UNLOCK	HMFUOFR	Unlock an Entity
VALUES	URMTH	URMTH	HPS User Relation Method
VIEW	ACTIONS	ACTIONS	List Entity Actions
VIEW	ADDE	ADDE	Add an Entity
VIEW	ADDR	ADDR	Add a Relation
VIEW	BE	BE	Browse Entity Attributes
VIEW	BK	HMFKEYB	Browse Keywords
VIEW	BR	BR	Browse Relation Attributes
VIEW	BTXE	TEXTBE	Browse Text for Entity
VIEW	BTXR	TEXTBR	Browse Text for Relation
VIEW	CO	HMFCOS0	Change Ownership
VIEW	DE	DELETEE	Delete an Entity
VIEW	DR	DELETER	Delete a Relation
VIEW	EW	HMFXTWU	Extended Where Used Report
VIEW	K	HMFKEY0	Define Keywords
VIEW	LE	LISTE	List Entity Instances
VIEW	LOCK	HMFUOLK	Lock an Entity
VIEW	LR	LISTR	List Relation Instances
VIEW	ME	MAINTE	Maintain Entity Attributes
VIEW	MR	MAINTR	Maintain Relation Attributes
VIEW	PR	PREPVIEW	Prepare View
VIEW	REFRESH	REFRESH	Refresh Entity
VIEW	REP	HMFREP	Produce Report

VIEW	RES	RESVIEW	View Results of View Actions
VIEW	SA	SAVEAS	Save As Current Entity
VIEW	SRCH	SEARCH	Search for Keywords
VIEW	TXE	TEXTE	Define Text for Entity
VIEW	TXR	TEXTR	Define Text for Relation
VIEW	UEMTH	UEMTH	HPS User Entity Method
VIEW	UNLOCK	HMFUOFR	Unlock an Entity
VIEW	URMTH	URMTH	User Relation Method
W			
WEAKENT	ACTIONS	ACTIONS	List Entity Action
WEAKENT	ADDE	ADDE	Add an Entity
WEAKENT	ADDR	ADDR	Add a Relation
WEAKENT	BE	BE	Browse Entity
WEAKENT	BK	HMFKEYB	Browse Keywords
WEAKENT	BR	BR	Browse Relation Attributes
WEAKENT	BTXE	TEXTBE	Browse Text for Entity
WEAKENT	BTXR	TEXTBR	Browse Text for Relation
WEAKENT	CO	HMFCOS0	Change Ownership
WEAKENT	DE	DELETEE	Delete an Entity
WEAKENT	DR	DELETER	Delete a Relation
WEAKENT	EW	HMFYWU	Extended Where Used Report
WEAKENT	K	HMFKEY0	Define Keywords
WEAKENT	LE	LISTE	List Entity Instances
WEAKENT	LOCK	HMFUOLK	Lock an Entity
WEAKENT	LOCKST	LOCKST	Get Lock Status
WEAKENT	LR	LLISTR	List Relation Instances
WEAKENT	ME	MAINTE	Maintain Entity Attributes
WEAKENT	MR	MAINTR	Maintain Relation Attributes
WEAKENT	REFRESH	REFRESH	Refresh Entity
WEAKENT	RES	RESGLBL	View Results of Actions
WEAKENT	SA	SAVEAS	Save As Current Entity
WEAKENT	SRCH	SEARCH	Search for Keywords
WEAKENT	TXE	TEXTE	Define Text for Entity
WEAKENT	TXR	TEXTR	Define Text for Relation
WEAKENT	UEMTH	UEMTH	User Entity Method
WEAKENT	UNLOCK	HMFUOFR	Unlock an Entity
WEAKENT	URMTH	URMTH	HPS User Relation Method
WINDOW	ACTIONS	ACTIONS	List Entity Actions
WINDOW	ADDE	ADDE	Add an Entity

WINDOW	ADDR	ADDR	Add a Relation
WINDOW	ASSIGNCU	HMFACU0	Assign Configuration Unit
WINDOW	BE	BE	Browse Entity Attributes
WINDOW	BK	HMFKEYB	Browse Keywords
WINDOW	BR	BR	Browse Relation Attributes
WINDOW	BTXE	TEXTBE	Browse Text for Entity
WINDOW	BTXR	TEXTBR	Browse Text for Relation
WINDOW	CO	HMFCOS0	Change Ownership
WINDOW	DE	DELETEE	Delete an Entity
WINDOW	DR	DELETER	Delete a Relation
WINDOW	EW	HMFYWU	Extended Where Used Report
WINDOW	K	HMFKEY0	Define Keywords
WINDOW	LE	LISTE	List Entity Instances
WINDOW	LISTRBD	LISTRBD	List Rebuild Contents
WINDOW	LOCK	HMFUOLK	Lock an Entity
WINDOW	LR	LISTR	List Relation Instances
WINDOW	ME	MAINTE	Maintain Entity Attributes
WINDOW	MR	MAINTR	Maintain Relation Attributes
WINDOW	PR	PREPWIN	Prepare Window
WINDOW	REFRESH	REFRESH	Refresh Entity
WINDOW	REP	HMFREP	Produce Report
WINDOW	RES	RESWIN	View Results of Window Actions
WINDOW	SA	SAVEAS	Save As Current Entity
WINDOW	SRCH	SEARCH	Search for Keywords
WINDOW	SUPERPR	SUPERPR	Super Prepare
WINDOW	TS	HMFTSID	Transaction Switch
WINDOW	TXE	TEXTE	Define Text for Entity
WINDOW	TXR	TEXTR	Define Text for Relation
WINDOW	UEMTH	UEMTH	User Entity Method
WINDOW	UNLOCK	HMFUOFR	Unlock an Entity
WINDOW	URMTH	URMTH	User Relation Method

Backwards Compatibility

Backwards Compatibility

In AppBuilder, there are some entities and relationships that appear on the mainframe but not on the workstation (PC). These entities serve to maintain backward compatibility with earlier versions of AppBuilder. Do not use these relationships or properties with newer versions of AppBuilder. They are listed here for informational purposes only.

- [Additional Entities](#)
- [Additional Relationships](#)

There are also some new workstation features that have not yet been implemented on the host. Consequently, there are some entities and

relationships that appear on the workstation, but not on the host.

Additional Entities

The supported platforms for each entity are outlined in the following sections:

Supported Platforms		
Component	Logical Process	Rule
Database Entity	Migration	Server
File	Project	Symbol
Group	Rebuild Package	Table
Help Text	Relationship	User

Component

The following relationships appear on the host, but not on the workstation:

- A Component accesses a Database
- A Component is-contained-in a Server
- A Component is-root-of Rebuild Package

Database Entity

The following relationships appear on the host, but not on the workstation:

- A Database is accessed-by a Component
- A Database is accessed-by a Rule

File

The following relationship appears on the host, but not on the workstation:

A File is-forwarded-from a File

Group

The following relationships appear on the host, but not on the workstation:

- Group is Profiled by Migration
- Group is profiled by Rebuild Package

Help Text

The following properties appear on the host, but not on the workstation:

- Country
- Language

Logical Process

The following property appears on the workstation, but not on the host:

Mode attribute called Logical Function

Migration

The following properties appear on the host, but not on the workstation:

- Description
- Data Set Prefix

Project

The following relationship appears on the host, but not on the workstation:Project is-member-of Group
The following properties appear on the host, but not on the workstation

- Project ID
- Description

Rebuild Package

This entity does not exist on the workstation.

Relationship

The following relationships appear on the host, but not on the workstation:

- A Relationship implements an Attribute
- A Relationship implements a Data Type
- A Relationship implements an Entity

Rule

The following relationship appears on the host, but not on the workstation:

A Rule accesses a Database

Server

The following relationship appears on the host, but not on the workstation:

A Server derives a Server

Symbol

The following properties are required on the host:

- Symbol define
- Symbol encoding

Table

The following relationship appears on the host, but not on the workstation:

A Table is-basis-of a Table

User

The following property appears on the workstation, but not on the host:

Project

The following properties appear on the host, but not on the workstation:

- Job account information
- Job name
- Job options

Additional Relationships

The relationship properties support for each platform is outlined in the following sections:

- [Is-carried-out-at](#)
- [Is-involved-in](#)
- [Is-modified-by](#)
- [Is-profiled-by](#)
- [Is-responsible-for](#)

- [Is-signed-by](#)
- [Is-site-of](#)
- [Is-supported-by](#)
- [Refines-into](#)
- [Replaces](#)

Is-carried-out-at

The following properties appear on the workstation, but not on the host:

- Major involvement in function
- Minor involvement in function

Is-involved-in

This relationship does not appear on the host.

Is-modified-by

This relationship does not appear on the host.

Is-profiled-by

The following property appears on the host, but not on the workstation:

Profile type

Is-responsible-for

This relationship does not appear on the host.

Is-signed-by

This relationship does not appear on the host.

Is-site-of

This relationship does not appear on the host.

Is-supported-by

This relationship does not appear on the host.

Refines-into

The following property does not appear on the host:

Condition flag

Replaces

This relationship does not appear on the host.