DOWNLOAD PDF OBJECT DATA MANAGEMENT

Chapter 1 : Object Data Management Group Definition from PC Magazine Encyclopedia

The Object Data Management Group (ODMG) was conceived in the summer of at a breakfast with object database vendors that was organized by Rick Cattell of Sun Microsystems.

Projects spatial data from one coordinate system to another. Also, you can use the Define Projection tool to permanently assign a coordinate system to the dataset. Coverages, VPF Coverages, raster datasets, and raster catalogs are not supported as input to this tool. Use the Project Raster tool to project raster datasets. When no geographic or datum transformation is required, no drop-down list will appear on the parameter and it is left blank. When a transformation is required, a drop-down list will be generated based on the input and output datums, and a default transformation will be picked. For a list of transformations and their area of use, see the following knowledge base article: When projecting the complex data types listed below, certain operations need to be performed on the resulting data: A feature dataset containing a network dataset: A feature dataset containing a topology: If the input participates in relationship classes as with feature-linked annotation , the relationship class will be transferred to the output. The exception to this rule relates to participating stand-alone tables. Features that fall completely outside the horizon will be written to the output with a Null shape. These can be deleted using the Repair Geometry tool. Feature classes participating in a geometric network cannot be projected independentlyâ€”the entire feature dataset containing the network needs to be projected. Many geoprocessing tools honor the output coordinate system environment setting, and in many workflows you can use this environment setting instead of using the Project tool. For example, the Union tool honors the output coordinate system environment setting, which means you can union several feature classes together, all of which are in a different coordinate system, and write the unioned output to a feature class in an entirely different coordinate system. Learn more about geoprocessing environments Selection and definition query on layers are ignored by this toolâ€”all features in the dataset referenced by the layer will be projected. If you want to project selected features only, consider using the Copy Features tool to create a temporary dataset, which will only contain the selected features, and use this intermediate dataset as input to the Project tool. When a feature class within a feature dataset is used as input, the output cannot be written to the same feature dataset. This is because feature classes within a feature dataset must all have the same coordinate system. In this case, the output feature class will be written to the geodatabase containing the feature dataset. The Preserve Shape parameter, when checked, creates output features that more accurately represent their true projected location. Preserve Shape is especially useful in cases where a line or polygon boundary is digitized as a long, straight line with few vertices. If Preserve Shape is not checked, the existing vertices of the input line or polygon boundary are projected, and the result may be a feature that is not accurately located in the new projection. These extra vertices preserve the projected shape of the feature. The Maximum Offset Deviation parameter controls how many extra vertices are added; its value is the maximum distance the projected feature can be offset from its exact projected location as computed by the tool. When the value is small, more vertices are added. Choose a value that suits your needs. For example, if your projected output is for general small-scale cartographic display, a large deviation may be acceptable. If your projected output is to be used in large-scale, small-area analysis, a smaller deviation may be needed. To perform a vertical transformation, check the optional Vertical parameter on the dialog. By default, the Vertical parameter is disabled and is only enabled when the input and output coordinate systems have Vertical Coordinate System, and the input feature class coordinates have Z values. Also, additional data Coordinate Systems Data setup needs to be installed on the system. When you select the output coordinate system, you will be able to choose both the geographic or projected coordinate system and a vertical coordinate system VCS. If the input and output vertical coordinate systems are different, an appropriate vertical and an optional geographic datum transformations is available.

Chapter 2 : Object Data Management Group | Revolvy

The Object Data Management Group (ODMG) completed its work on object data management standards in and was disbanded. The final release of the ODMG standard can be found in the book "The Object Data Standard ODMG ", including the overall data model, and the Smalltalk and C++ bindings for ODMG.

Another group of object databases focuses on embedded use in devices, packaged software, and real-time systems. Technical features[edit] Most object databases also offer some kind of query language , allowing objects to be found using a declarative programming approach. It is in the area of object query languages, and the integration of the query and navigational interfaces, that the biggest differences between products are found. Access to data can be faster because an object can be retrieved directly without a search, by following pointers. Another area of variation between products is in the way that the schema of a database is defined. A general characteristic, however, is that the programming language and the database schema use the same type definitions. Multimedia applications are facilitated because the class methods associated with the data are responsible for its correct interpretation. An object can be viewed as the set of all its versions. Also, object versions can be treated as objects in their own right. Some object databases also provide systematic support for triggers and constraints which are the basis of active databases. The efficiency of such a database is also greatly improved in areas which demand massive amounts of data about one item. Standards[edit] The Object Data Management Group was a consortium of object database and object-relational mapping vendors, members of the academic community, and interested parties. Its goal was to create a set of specifications that would allow for portable applications that store objects in database management systems. It published several versions of its specification. The last release was ODMG 3. By , most of the major object database and object-relational mapping vendors claimed conformance to the ODMG Java Language Binding. Compliance to the other components of the specification was mixed. As a result, the ODMG disbanded in Many object database ideas were also absorbed into SQL: In Cook, Rai, and Rosenberger proposed to drop all standardization efforts to introduce additional object-oriented query APIs but rather use the OO programming language itself, i. NET, to express queries. As a result, Native Queries emerged. The work of the ODBT WG was suspended in March when, subsequent to the economic turmoil in late , the ODB vendors involved in this effort decided to focus their resources elsewhere. XQuery uses XML as its data model. Some of the ideas developed originally for object databases found their way into XQuery, but XQuery is not intrinsically object-oriented. Because of the popularity of XML, XQuery engines compete with object databases as a vehicle for storage of data that is too complex or variable to hold conveniently in a relational database. XQuery also allows modules to be written to provide encapsulation features that have been provided by Object-Oriented systems. Comparison with RDBMSs[edit] An object database stores complex data and relationships between data directly, without mapping to relational rows and columns, and this makes them suitable for applications dealing with very complex data. Pointers are linked to objects to establish relationships. Another benefit of an OODBMS is that it can be programmed with small procedural differences without affecting the entire system.

Chapter 3 : Projectâ€”Data Management toolbox | ArcGIS Desktop

In addition to an Index, the book includes an Annotated Bibliography of reference materials covering seminal papers in object data management, new object data management systems, and important background material in programming languages and traditional database technology.

Contributors In this article This topic describes how you can use the data management framework to manage data entities and data entity packages in Microsoft Dynamics for Finance and Operations. The data management framework consists of the following concepts: A data entity is a conceptual abstraction and encapsulation of one of more underlying tables. A data entity represents a common data concept or functionality, for example, Customers or Vendors. Data entities are intended to be easily understand by users familiar with business concepts. A project that contains configured data entities, which include mapping and default processing options. A job that contains an execution instance of the data project, uploaded files, schedule recurrence , and processing options. Histories of source to staging and staging to target jobs. A single compressed file that contains a data project manifest and data files. This is generated from a data job and used for import or export of multiple files with the manifest. The data management framework supports using data entities in the following core data management scenarios: Data migration Set up and copy configurations Integration Data entities Data entities provide conceptual abstraction and encapsulation of underlying table schema that represent data concepts and functionalities. In Microsoft Dynamics AX , most tables, like the Customer and Vendor tables, were de-normalized and split into multiple tables. Data entities were introduced as part of data management to be used as a layer of abstraction to easily understand by using business concepts. The concept of data entities combines those different concepts into one. The following table shows core data management scenarios. Data Migration Migrate reference, master, and document data from legacy or external systems. Data migration Using the data management framework, you can quickly migrate reference, master, and document data from legacy or external systems. The framework is intended to help you quickly migrate data by using the following features: You can select only the entities you need to migrate. If an import error occurs, you can skip selected records and choose to proceed with the import using only the good data, opting to then fix and import the bad data later. You will be able to partially continue and use errors to quickly find bad data. You can move data entities straight from one Finance and Operations system to another, without having to go through Excel, or XML. Data imports can be easily scheduled using a batch, which offers flexibility when it is required to run. For example, you can migrate customer groups, customers, vendors, and other data entities in the system at any time. Set up and copy configuration You can use the data management framework to copy configurations between companies or environments, and configure processes or modules using Microsoft Dynamics Lifecycle Services LCS. The data management framework allows you to: Move data between two similar systems Discover entities and dependencies between entities for a given business process or module Maintain a reusable library of data templates and datasets Use data packages to create incremental data entities. Data entities can be sequenced inside the packages. You can name data packages, which can be easily identifiable during import or export. When building data packages, data entities can be mapped to staging tables in grids or by using a visual mapping tool. You can also drag-and-drop columns manually. View data during imports, so you can compare data, and ensure that it is valid. Working with data entities The following sections provide quick snapshots of the different functionalities of data management using data entities. The goal is to help to you strategize and make effective decisions on how to best utilize the available tools during data migration. You will also find tips and tricks on how to effectively use each area during data migration. A list of available data entities for each area can also be found with the suggested data sequences, showing data dependencies. The information in this document can be used as a guide for creating your own packages. The description of each data entity shows what the object contains and if it is needed during data migration. Sequencing There are two types of sequencing that should be considered when working with data entities. Sequencing data entities within a data package Sequencing the order of data package imports Sequence data entities within a data packages When a user adds data entities to a data project, by

default, a sequence is set for the order in which the entities will load. The first entity added to the project will be set as the first entity to load, the next entity added will be second, the next entity will be third, and so on. For example, if a user added two entities in this order, Sales tax codes and Sales Tax groups, then Sales tax codes is assigned an entity sequence of 1. The sequence level indicates that the second entity will not start the import process until the first level is finished. To view or edit a sequence, click the Entity sequence button on the Action Pane of the data project. In the Definition group entity sequence, you can see the execution units and the sequence. You can change sequence by selecting the data entity in the list, setting a different Execution unit or Sequence in level, and then clicking Update selected. After clicking Update selected, the entity will move up or down in the entity list. Example The following screenshot shows the entity sequence that is set for the Sales Tax CodeGroups data package. In order to successfully import sales tax codes and groups, the sales tax codes and details have to be loaded first, before sales tax groups can be imported. Other related sales tax entities that are not dependent upon other data entities being loaded are included in the package. This data entity will start loading immediately because there are no dependencies on other entities loading before it. The numbering format that has been created for the data packages within LCS are as follows: Data type setup, master, transaction Third segment: Sequence number The following tables provide more information about the default numbering format. Module numbers Data type numbers Sequence number Data packages follow the sequence number, followed by the module abbreviation, and then a description. The following example shows General ledger data packages. Mapping When working with data entities, mapping an entity to a source is automatic. The automatic mapping of fields can be overridden if needed. View mapping To view how an entity is mapped, locate the tile for the entity in the project, and then click View map. We provide mapping visualization view default and mapping details view. These fields must be mapped in order to work with the entity. Other fields can be unmapped as required when working with the entity. To unmap a field, highlight the field in either column Entity or Source , click Delete selection, and then click Save. After saving, close the form to return to the project. The field mapping from source to staging can also be edited after import using the same process. Regenerate a map If you have extended an entity added fields or if the automatic mapping appears to be incorrect, the mapping of the entity can be regenerated in the Mapping form. To do this, click Generate source mapping. A message will display asking, "Do you want to generate the mapping from scratch? Generate data If you have fields in entities that you want the system to generate data for on import, instead of providing the data in the source file, you can use the auto-generated functionality in the mapping details for the entity. For example, if you want to import customers and customer address information, but the address information was not previously imported with the Global Address Book entities, you can have the entity auto-generate the party number upon import and the GAB information will be created. To access this functionality, view the map of the entity and click the Mapping details tab. Select the fields that you want to auto-generate. This will change the source field to Auto. Turn off automatically generated number sequences Many entities support automatic generation of identifiers based on number sequence setup. For example, when creating a product, the product number is automatically generated and the form does not allow you to edit values manually. It is possible to enable manual assignment of number sequences for a specific entity. After you have enable manual assignment, you can provide manually assigned numbers instead. Export Export is the process of retrieving data from a system using data entities. The export process is done through a project. When exporting, you have a lot of flexibility as to how the export project is defined. You can choose which data entities to export, but also the number of entities, the file format used there are 14 different formats to choose for export , and apply a filter to each entity to limit what is exported. After the data entities have been pulled into the project, the sequencing and mapping described earlier can be performed for each export project. After the project is created and saved you can export the project to create a job. During the export process, you can see a graphical view of the status of the job and the record count. This view shows multiple records so you can review the status of each record prior to downloading the actual files. While importing the system users entity, you may receive an integrity violation error if there is a guest user in the exported package. The guest user must be deleted from the package in order for the entity to work. If a record already exists in the UserInfo table the Admin record would most likely always exist , the import will fail for those

records but work for other records. After the job is completed you can choose how to download the files: If there are multiple data entities in the job, choosing the package option will speed up the upload process. The package is a zip file, containing a data file for each entity as well as a package header and manifest. These additional documents are used when importing in order to add the data files to the correct data entities and sequence the import process. Import Import is the process of pulling data into a system using data entities. The import process is done through the Import tile in the Data Management workspace. Data can be imported either for individual entities or for a group of logically related entities that are sequenced in the correct order. The file formats vary depending on the type of import. For an entity, it can be an Excel file that is comma-separated, tab-separated, or text. For a data package, it is a.

Chapter 4 : Object Data Management Group - Wikipedia

Object Model. This was based on the Object Management Group's Object Model. The OMG core model was designed to be a common denominator for object request brokers, object database systems, object programming languages, etc.

What is a Customer? Data Objects Data Standards and Master Data Management You can only get the most out of your data warehouse if you have a uniform understanding of your data. Such fields as "customer" must mean the same to all business units. We explain what data standards are and why they are key to your success. By David Loshin April 20, I often pose a question to our clients: As a consequence, the semantics associated with those replicated data objects have also diverged. Here is an example: The first two are populated with real addresses, but apparently the third address field was never used for addresses. In fact, closer inspection reveals similar notes embedded within the name fields as well. A straightforward name-and-address cleaning is likely to remove the extraneous data in the name fields and present the customer notes in the address fields as invalid addresses. In this case, cleansing the names and addresses for the master file will introduce flaws into the business process as that extraneous, yet meaningful, data is eliminated. The issue here is not the mechanics of integration. Having identified those data objects my favorite term , the goal is to determine the semantics assigned to those objects within each source application as a prelude to the integration process. And this is where data standards and data governance come in; these will help prevent the unending conflicts on data terms and their definitions. A data standards process is an approach to synchronizing the various metadata aspects of shared or exchanged data objects. By formalizing the process of gaining consensus among the different participants, and enabling their active engagement in both defining and governing that process, we can evolve a collection of well-defined business terms, information object models, information exchange packages, and a means for mapping these shared object definitions into the models that are ingrained within the legacy environment. In addition, a data standards process can be used to help harmonize common business language terms and data elements to represent those terms as part of a master data management program. What is a data standard? An agreement between parties on the definitions of common business terms and the ways those terms are named and represented in data; A set of rules that may describe how data is stored, exchanged, formatted, or presented; A set of policies and procedures for defining rules and reaching agreement. But the essence of a standard is not just the consolidation of metadata, the formal description framework, or even the rules for definition and governance. Instead, it is the premise that: The individual stakeholders desire to work together to develop a common language; Participants are provided with an opportunity to participate in the process through proposing new standards, evaluating proposals, and providing comments; Most importantly, the participants agree that the defined and agreed-to standard is the single set of guidelines to be used for data integration and sharing. This leads to the following challenges: Absence of clarity for object semantics. Relying on the implied meanings associated with business terms may be fine when a system is self-contained, but as soon as there is a need to compare values between two or more environments, subtle differences in meanings become magnified. The ambiguity is typically aligned along application, and subsequently, departmental lines. Exposing ambiguity will encourage individuals to promote their own semanticsâ€”to the exclusion of others. This plants the seeds for organizational conflict. People tend to be less than precise in standard conversations, because humans can derive understanding through context. However, in an imprecise environment, it is difficult to resolve measurements and metrics into a unified view. Variance in source systems. Aside from the semantics issues, implementation decisions may create reliance on application frameworks, leading to religious wars e. Flexibility of storage and exchange mechanisms. The multiple modes by which data is exchanged can expose conflicts in metadata descriptions when trying to create a seamless means for integration. This may mean creating adapters that can transform data objects between formats, such as between records in flat files and XML documents. As an example of one of these types of challenges, one question that might arise when attempting to integrate data values from systems with different constraints into a master system has to do with value types and sizes. For example, when merging product names, application A may use a fixed-size field, while application B may allow for variant-length

names. When aggregating product data, it might be reasonable to assume the fixed-size format to ensure that no subsequently shared data exceeds the storage space allowed within application A. On the other hand, it might be reasonable to assume the variant-length size, to accommodate application B. The issue occurs when the master copy of the data is published back outâ€”either approach introduces inconsistencies between the master table and the applications. For the purposes of master data integration, the objective of a data standards approach is to identify all the objects and to reach consensus on their respective definitions, types, formats, and structures. While the approaches that different organizations take to reach consensus may differ, here are some suggestions for getting started: A free-form brainstorming session where the terms used are identified, documented, and made available for discussion of a more precise definition. A gap analysis to determine whether definitions already exist, and if not, what additional information is needed. At the same time, the terms that are discovered can be categorized within some logical hierarchy. Researching the information needed based on the previous step to locate complete and acceptable metadata definitions and specifications, if they exist. For each term, seek consensus by reviewing all the distinct definitions for each term, and either accepting a definition, revising a definition, or creating a new definition. As the participants reach agreement, the definitions are documented along with any supporting information that influenced the decision. The definitions and supporting information are assembled in a metadata repository. There is great value in establishing a governance framework for developing master data semantics, and when embarking on a master data integration project, it is wise to incorporate a data standards program into the process. David Loshin is the president of Knowledge Integrity, Inc, a consulting and development company focusing on customized information management solutions. Along with consulting on numerous data management projects over the past 20 years, David is also a prolific author regarding business intelligence best practices; he has written numerous books and papers on data management. David is a frequent invited speaker at conferences, online seminars, and sponsored websites and channels.

Chapter 5 : OMG | Object Management Group

The Object Management GroupÂ® (OMGÂ®) is an international, open membership, not-for-profit technology standards consortium. Founded in , OMG standards are driven by vendors, end-users, academic institutions and government agencies.

Chapter 6 : What is a Database Object? - Definition from Techopedia

The Master Data Management and Data Loader are not intended for managing object types managed in an ERP system. In a Nutshell The Master Data Management module includes several tabs used to create and manage specific data object types.

Chapter 7 : ODMG Standard - calendrierdelascience.com

An object-oriented database management system (OODBMS) is a database management system that supports the creation and modeling of data as objects. OODBMS also includes support for classes of objects and the inheritance of class properties, and incorporates methods, subclasses and their objects.

Chapter 8 : Azure Storage - Secure cloud storage | Microsoft Azure

Binds WMI class information to the management object. Get Populates a SerializationInfo with the data necessary to deserialize the field represented by this instance.

Chapter 9 : Data management - Finance & Operations | Dynamics | #MSDynFO | Microsoft Docs

DOWNLOAD PDF OBJECT DATA MANAGEMENT

An object database is a database management system in which information is represented in the form of objects as used in object-oriented programming. Object databases are different from relational databases which are table-oriented.