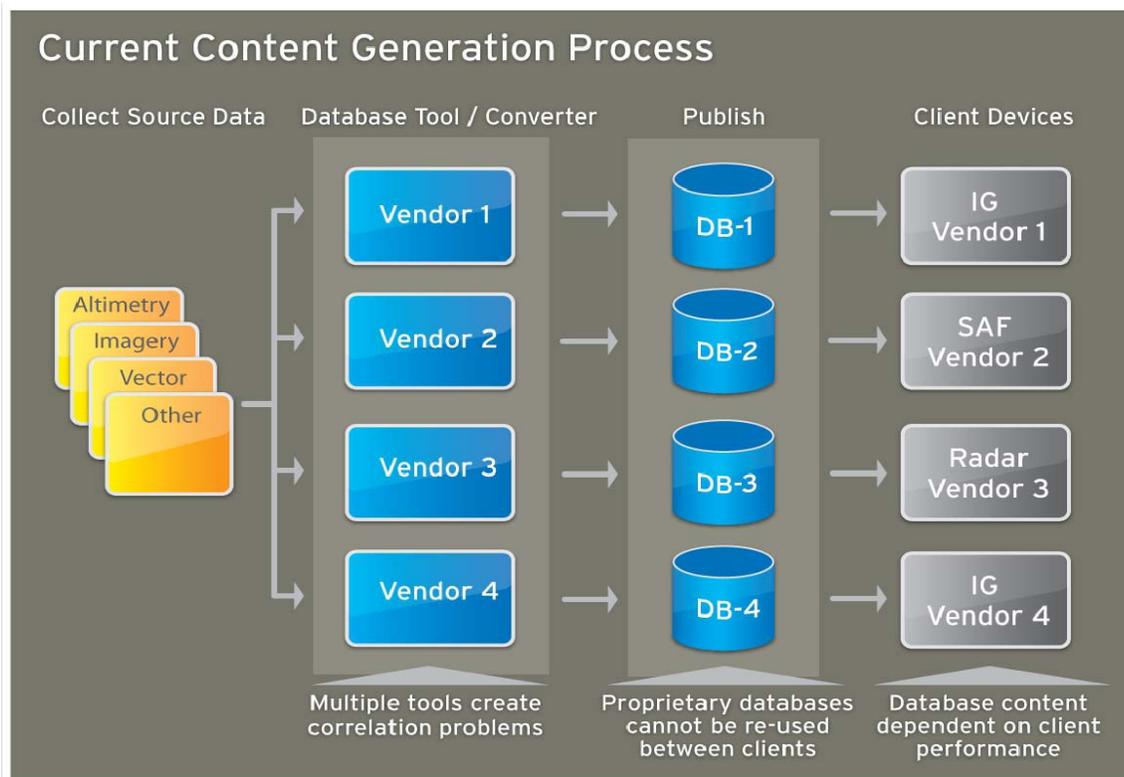


# COMMON DATABASE (CDB)

Traditional synthetic environment data base generation techniques until now, were mainly processed using an offline approach and were customized to the specific runtime system with which they needed to work. This process is shown in figure 1. The offline approach, while necessary because of the proprietary nature of the runtime system, was a daunting process for larger simulations, often requiring hours and even days of processing before the database could be published to the required format(s). Correlation errors resulting from multiple compilation/publishing tools were frequent, and updates or changes were often difficult and very time consuming to implement. This laborious process needed to be repeated for every runtime instance, including visual system, SAF, and sensor, as each runtime system required its own unique database. In the end, compromises were made that reduced the level of portability, interoperability, scalability, abstraction, and correlation of the database with other simulator client-devices.

As a result of these database and database processing issues, applications became more difficult and more costly to create and maintain. And, as the end user's requirements have broadened and have become more demanding and specific, these issues have become exacerbated. Most database formats in use today still require a full offline recompilation of the database specific to each runtime client.



The **Common Database (CDB)** specification is an open synthetic environment database specification. It is intended as a simulation specification for use in producing a unified synthetic representation of the world. A database built to the CDB specification is referred to as a Common Database (CDB) and is a single-copy data repository from which various simulator client-devices are able in real-time to simultaneously retrieve relevant information to perform their respective runtime simulation tasks. The CDB specification offers a database structure that is well suited as an offline repository for the database authoring tools. A CDB is plug-and-play interchangeable between workstations whose authoring toolset conforms to the CDB specification. It is also appropriate as an online (or runtime) repository for simulators that conform to the CDB specification. Since the CDB can be used directly by some or all of the simulator client devices, it is

considered to be a runtime environment database. CDB supports all of the most common features required in a synthetic environment, including imagery, altimetry, 3D culture, sensors, bathymetry, light points, seasonal effects, model skins, and damaged representations.

CDB is unique in that it is an open database format, a source data repository, and a runtime publishing format all in one. As an open commercial standard, CDB meets the needs of the industry as a whole because

- The standard is improved over time with input from the development community.
- Lessons learned can be—and currently are—incorporated into the format.
- Users may choose to add support for CDB within their own systems and can then benefit from those advances.
- Databases conforming to the standard can be reused as well as interchanged and shared between end users with CDB-compliant runtime systems.
- Redundant implementation investigation work is either eliminated or reduced.

By following the standard, systems integrators have, from the very beginning, a clear and consistent path for developing powerful synthetic environments. As a source data repository, the native format for CDB is comprised of the most commonly used source data formats, including TIFF, GEO-Tiff, OpenFlight, Shapefile, and XML; therefore, there is no 'conversion' of data. This greatly speeds up the database creation and publishing process and also allows for the rapid modification of databases. CDB is both a source data repository as well as a runtime format. It is designed to be read by all of the possible runtime systems, including—but not limited to—visual systems, Navaids, radar, and SAFs, meaning that changes in the database source format are immediately represented in the runtime systems. And, because these runtime systems all use the same database, they are naturally correlated.

