

WISE CONNECTIVITY

With our products CoDE and CoRE, you can achieve connectivity – regardless of protocols, standards and architectures. It is simply done by configuration.

FEATURES

- Connectivity – regardless of protocols, standards and architectures
- Information interoperability – achieved by configuration
- “Information centric integration”
- Digital signing of drivers, to assure information security
- Digital signing of connectivity solutions, to assure information security
- Local connectivity solutions
- Distributed connectivity solutions
- User-defined connectivity solutions
- Recording the information transactions in the connectivity solution

BENEFITS

Support for protocols:

- HLA 1.3 and HLA 1516,
- DIS
- Database (MS SQL Server, MySQL)
- RTPS
- MIP (C2IEDM)
- Google Earth
- Link 16
- Web Services
- and more protocols

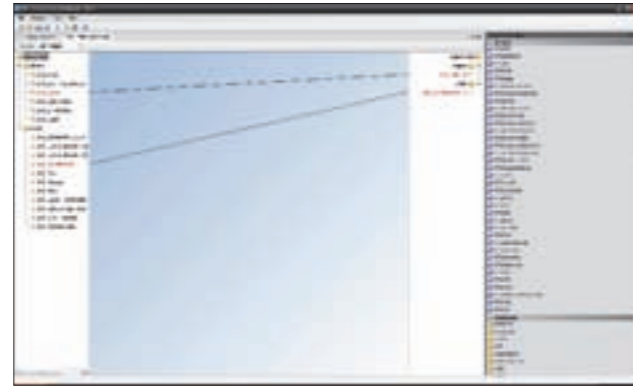
- Use connectivity drivers to access systems as building blocks
- Software Developer Kits to support 3rd party development and enhancements the SDKs are compatible with MS Visual Studio 2005 SP1 and uses wizards to create new projects fast and easy
- WISE Driver SDK for 3rd party development of connectivity drivers
- WISE Transformer SDK for 3rd party development of connectivity transformers
- WISE Service SDK for 3rd party development of connectivity services.
- WISE Application SDK for 3rd party development of WISE Applications



INTEGRATION BY CONFIGURATION HERE IS HOW:



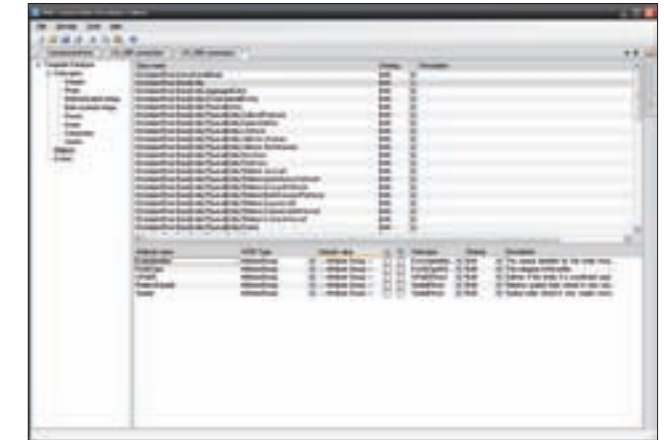
Connectivity View. Here you select which systems to integrate; just drag-and-drop their drivers.



Connection View. Then you specify how information should be related between different systems.



Coupling Set View. Then you define how the related information need to be transformed to be consistent.



Template Database View. Here you can view and manage the information from a system.

Configure the information flow

Each application has a specific framework within the common environment. This framework communicates with “its own” application, and the information flow between the two is persistent. As soon as the information model in the application is updated, the information model in the framework is also updated – and vice versa. To make information flow within the infrastructure, you also need to set up connections between the different information models. More specifically, you need to specify how information is translated from one information model to another. As part of the configuration, you can also set up how fast information is to be updated between the common information model and an application specific information model, and vice versa.

Configure the possibilities

The data management functionality gives you the possibility to filter and transform the data before it is transferred. For instance, you can limit the number of entities transferred to and from an application; choose to only transmit entities in a certain area; and change the update rate of entities for slower applications.

Configure the communication drivers

The configuration tool can be delivered with a set of communication drivers. You can also develop your own drivers, for instance if you want to integrate a legacy application that is not supported by the drivers that come with WISE. Another reason to develop your own drivers could be the need to use a special encryption protocol. But in general you only have to select the relevant drivers, and specify communication parameters.

Configure the information models

The configuration tool is delivered with a set of information models. If needed, you can also create new information models within the configuration tool. To create a new information model, you specify relevant objects and their attributes. Normally, you only need to select relevant information models, and possibly specify some additional parameters.

Configure the connections

It usually requires some thinking and effort when setting up the connections between information models. You must specify how object attributes in one information model should affect object attributes in the other information model. You will also have to define when and how often the information should flow through the connections, every time something changes or periodically at some time.

CODE – Connectivity Designer Edition

The Connectivity Designer Edition (CoDE) is a graphical tool to create and manage connectivity solutions, connections, coupling and object models. With it, you can specify the required integration environment and its information infrastructure. Several configurations can be set up and saved in parallel to a library, and to develop environments tailored to a certain range of applications – just configure the information infrastructure. The user has access to the library with supported connectivity drivers and object models. When creating a connection, the user adds necessary transformers to synchronise the object models. All supported transformers could be used by drag-and-drop from the toolbox, and if you need to develop a new transformer, then use the WISE Connectivity Transformer SDK.

CORE – Connectivity Runtime Edition

The Connectivity Runtime Edition (CoRE) controls and executes the connectivity solution. Here you can load, start and stop the configurations that are created earlier. You can also log everything that happens in the common object model to file. The loaded drivers and their status are also shown, giving the user an easy overview of the running connectivity solution. The CoRE also features a log window where all the important information from the system and loaded drivers is published. Through debugging, you can have even more information in the log window, for instance to help you to quickly sort out a problem. With CoRE, you can easily switch between different environments that are tailored to a certain range of applications, and load your preferred configuration. No additional software programming is needed.

Connectivity Test Tool

The Connectivity Test Tool connects to a running CoRE and is used to inspect the objects the connected drivers has in their databases. It is also possible to create and destroy object in the databases or update the attributes of an object. Besides objects it is possible to create events and all events are logged for easy inspection. Objects in a database can be stored for later retrieval; this is very useful for testing.