RapidGen Genius Suite
Decision Model Execution Engine

Organizations that automate their decision-based business processes using decision modelling can:

- **Lower time to market** by allowing business analysts to build, review, test, and improve transparent decision-making models without specialist infrastructure or heavy IT involvement.
- **Lower costs** and **improve reliability** by automating the processes, by avoiding logical flaws, and by quickly identifying the required input data, ensuring there are no omissions or expensive redundancies.
- **Improve agility** by allowing the full impacts of proposed changes to be seen before implementation and by facilitating rapid amendments.

Decision modelling software is widely used to design the decision logic and structure. But many of the benefits are lost if decision models are **manually** turned into code:

- Manual programming can often introduce subtle errors.
- It takes longer to make changes and can prevent rapid development.
- Business analysts cannot read the code, reducing transparency, accuracy, and trust.
- Manually-written code can become inefficient and difficult to maintain.
- It is much harder to maintain traceability between manually produced code and the decision model.
- Over time, manually maintained code can become inconsistent with the decision model yielding a false impression of the organization's decision-making.

RapidGen Genius Suite completes the automation of decision-based processes in two steps, by automatically:

1 Generating Executable Code from Decision Models

- Importing DMN-compliant models from your decision modelling software.
- Translating and compiling these models into executable code.

2 Executing Decisions Ultra-Fast

- Accessing required data from data sources and executing decision logic.
- Processing data at high throughput and exporting results.
- Providing traceable and auditable records of the decision processes.
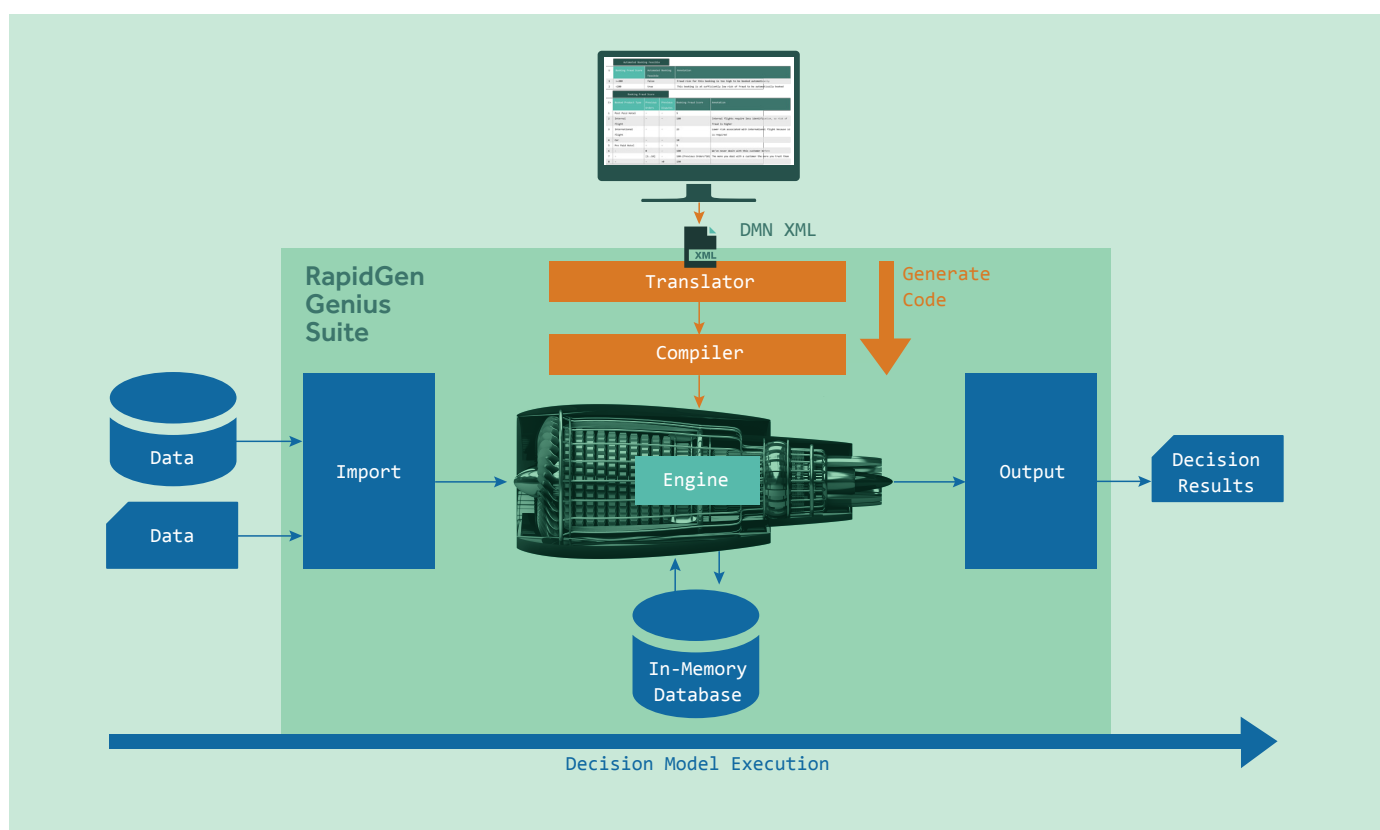
Generating Code

## Create executable code from your decision model in 4 easy steps

1. Design your decision tables in your chosen decision modelling software and export as DMN XML.

2. Import into RapidGen Genius Suite. This will automatically translate into RapidGen's own language, RPL.

3. Link DMN variables to data sources and parameters.

4. Run the RapidGen compiler to create highly-efficient executable code.

DMN

The Decision Model & Notation standard (DMN) is widely supported by decision modelling software vendors and the decision management community as a precise, code-free way to represent business logic.

Decision models can be passed seamlessly between software from different vendors that support this standard.

RapidGen Genius Suite can import DMN XML from the leading decision modelling tools.

Generate Code

DMN XML

Decision Table Import

Translator

RPL

Compiler
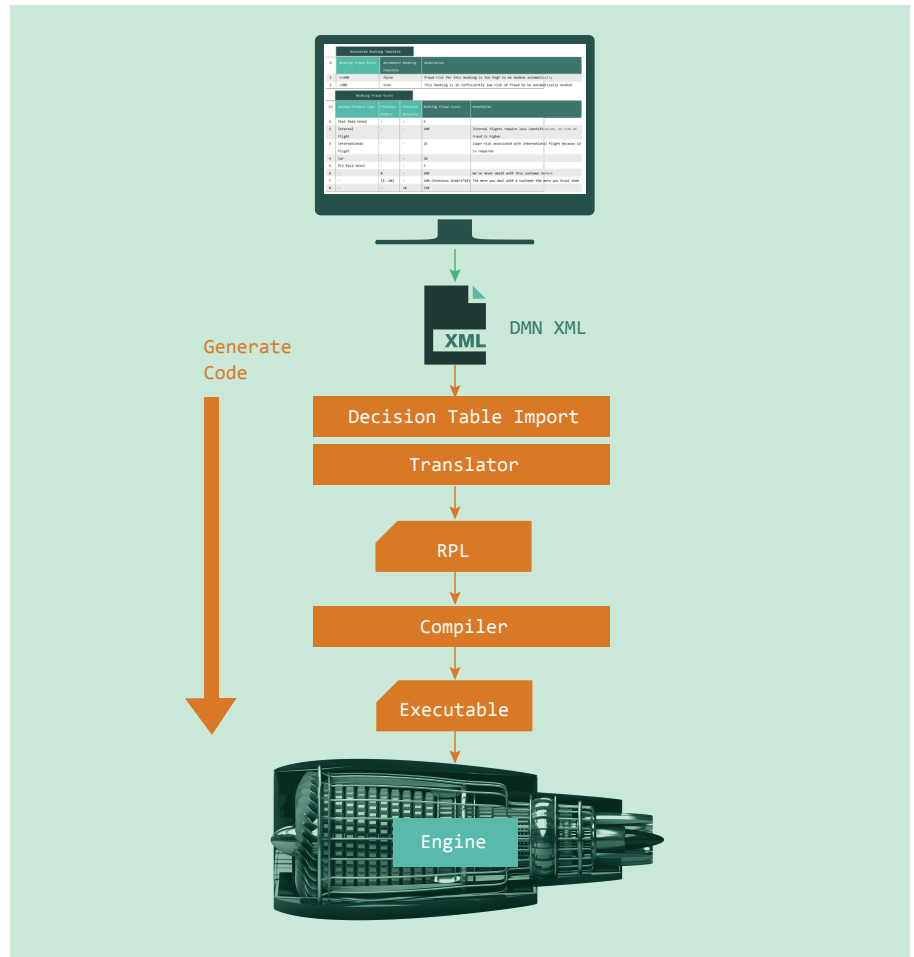
Executable

Engine

RPL

RPL is RapidGen's proprietary language, developed specifically to process decision tables.
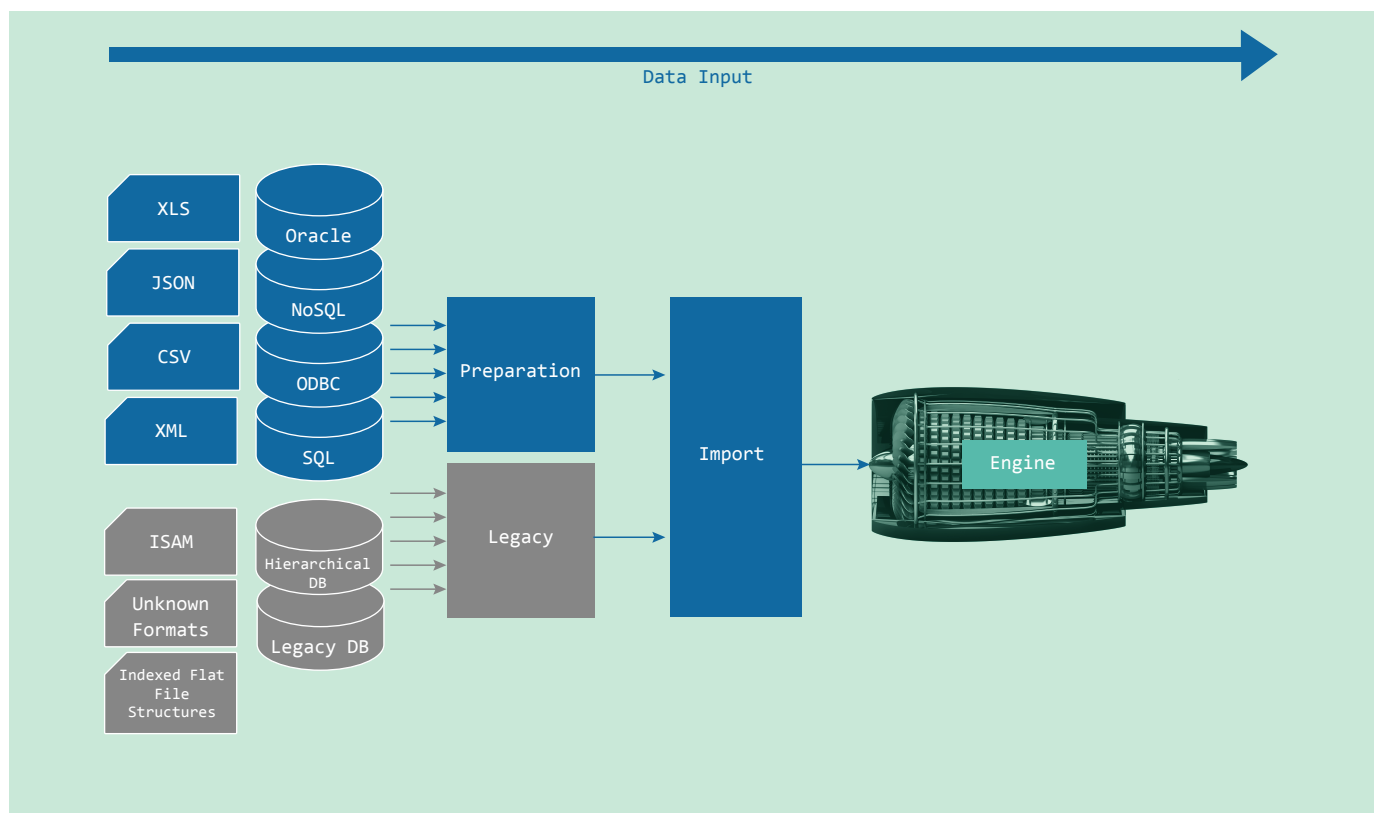
Decision Model is Primary

The decision model is the definitive version of the decision logic.

Running Decision Model Execution Engine

Having generated the executable code, you can start the execution engine. The engine processes data in 3 stages

1. Data Input   2. Execution   3. Data Output

1. Data Input

First, data is obtained from a wide range of heterogeneous data sources and formats.

Then, data is prepared for the execution engine
- Merging, concatenating, cleansing
- Sorting, indexing, deduplication

What-If Modelling

Data sources and decision models can be selected dynamically for what-if scenarios.

Binary Formats

Native support for data in binary formats, avoids the need for conversion and loss of precision.

RapidGen Legacy Module

Adding this module gives comprehensive support to a wide range of legacy:
- Architectures
- Data sources
- Data types
- Languages
- Operating systems

RapidGen consultants can even create bespoke interfaces for undocumented data sources

2. Execution

Each executable is compiled for optimal throughput and efficiency on the specific operating system.

- Most conditions and actions are performed inline.
- Decision table rules are stored in a machine register and manipulated using bitwise logic instructions for maximum efficiency.
- Data source access uses native drivers where available and optimized bulk throughput routines.
- Logic execution is fully scalable and unconstrained by data volume and quantity.

Decision Model Execution

Parameters

Executable

Import

Engine

Output

In-Memory Database
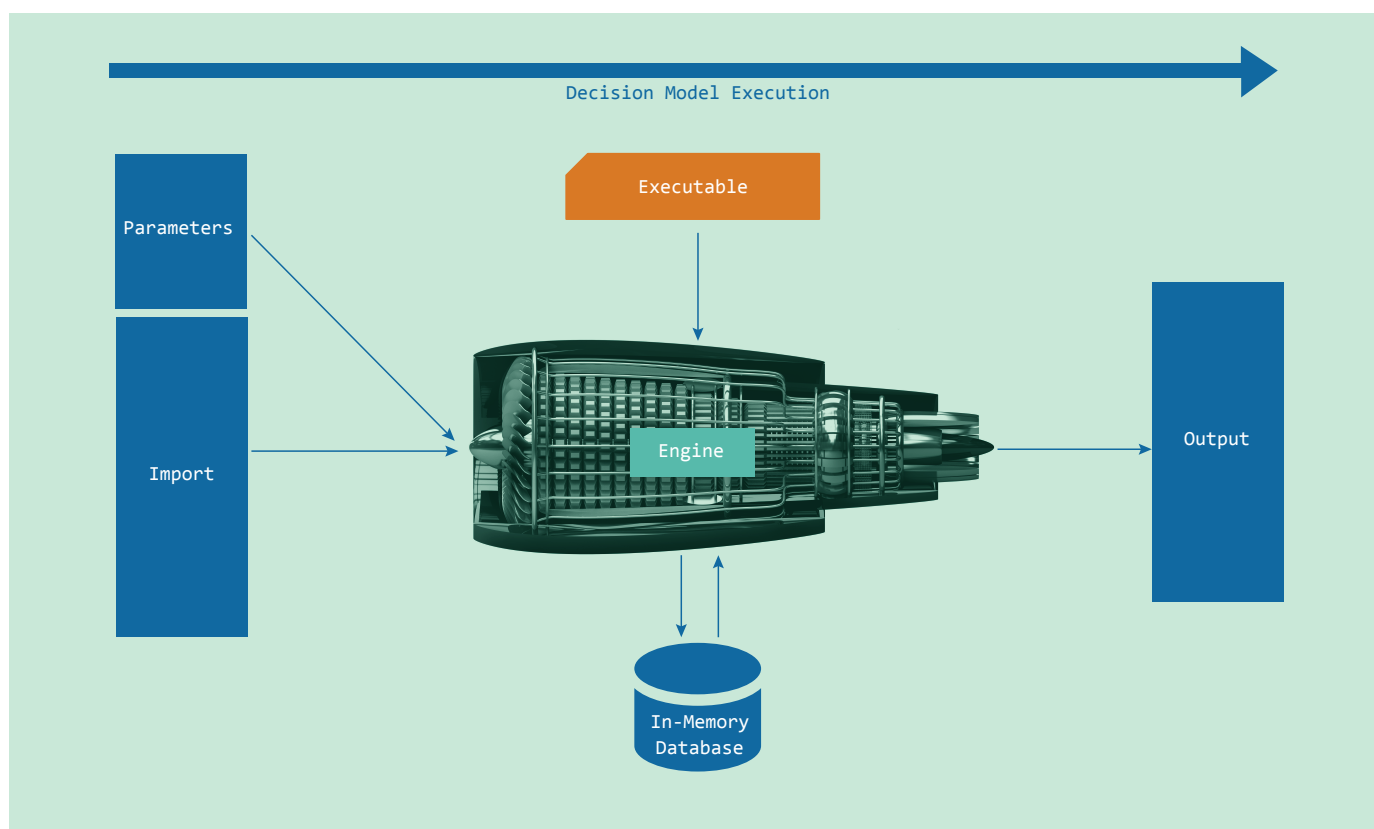
Traceability

Full traceability and auditable logging are optionally available.

Portability

The execution engine can run on
- Linux / Unix
- Windows
- 32-bit and 64-bit architectures

With the additional legacy module the engine can run on a variety of legacy minicomputers.

In-Memory Database

Disk-based I/O is minimized using an in-memory database
- Intermediate results
  e.g. prior to aggregation
- Inter-process data exchange

Shared access to in-memory database allows efficient multiple scenario processing of decision models.

3. Data Output

The engine produces output in standard formats, ready for

- Importing into decision modelling tools.
- Reporting and analysis.
- Integrating into other business processes.

Data Output

Engine → Output → XML / CSV / XLS / JSON / HTML
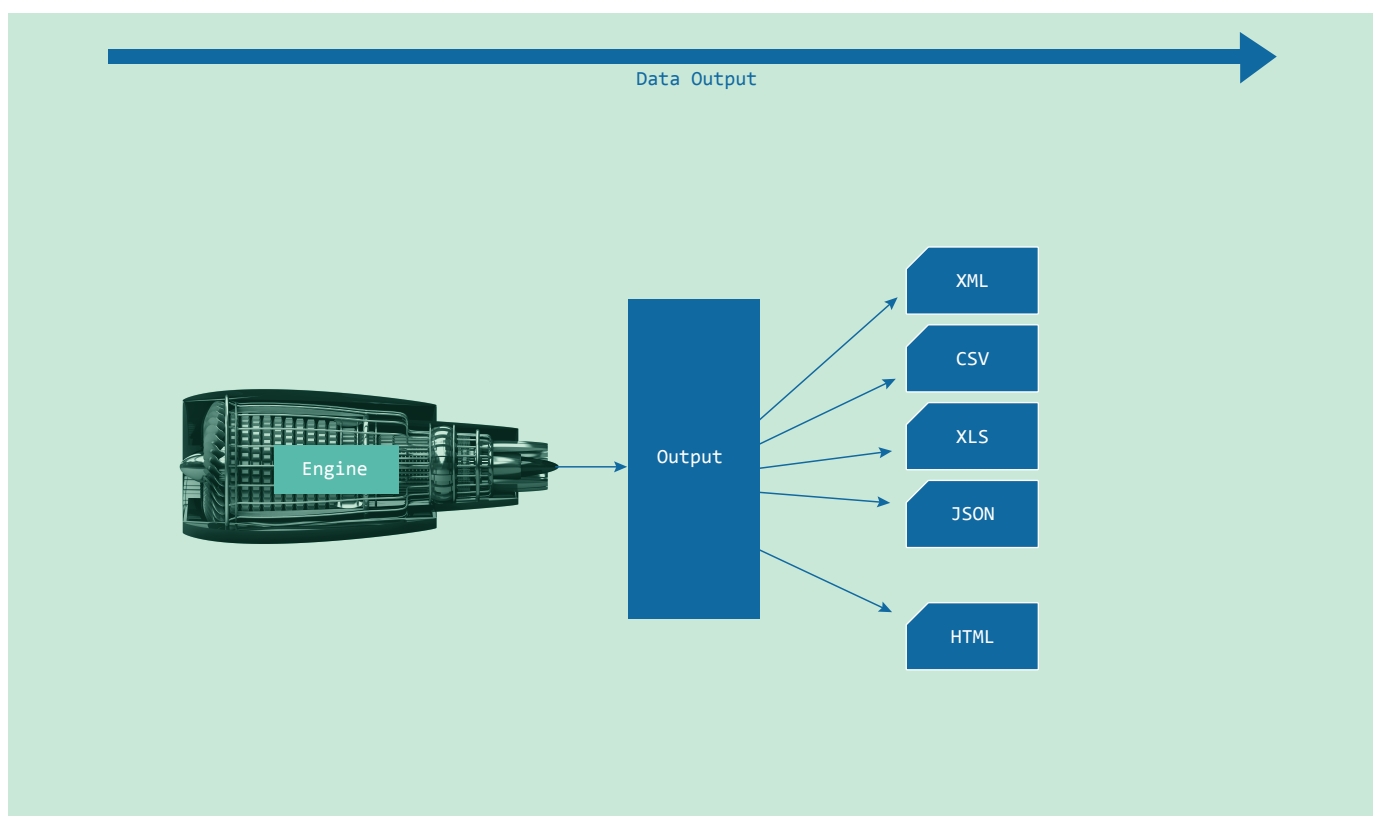
Decision outcomes can be exported in a range of standard output formats.

Exporting processing includes
- SQL-like operations
- Filters
- Sorting
- Aggregation
- Summary reports

Data Preview

Use lightweight webpages for simple interactive reporting.

Useful for
- Quick access to results
- Rapid prototyping
- Monitoring

RapidGen consultants can even create bespoke interfaces for undocumented data sources

Traceability

Show details of how decisions were reached
- Processing steps
- Which rules were fired

Decision Model Execution Engine

Hundreds of companies in dozens of industries around the world have been using RapidGen software to automate their decision-based business processes

- Accurately and transparently translating decision logic into code.
- Rapidly and flexibly implementing processes without manual coding.
- Generating executable code that is ultra-fast and scalable.
- Importing data from a wide range of data sources.