B4P

Beyond Former Performance.

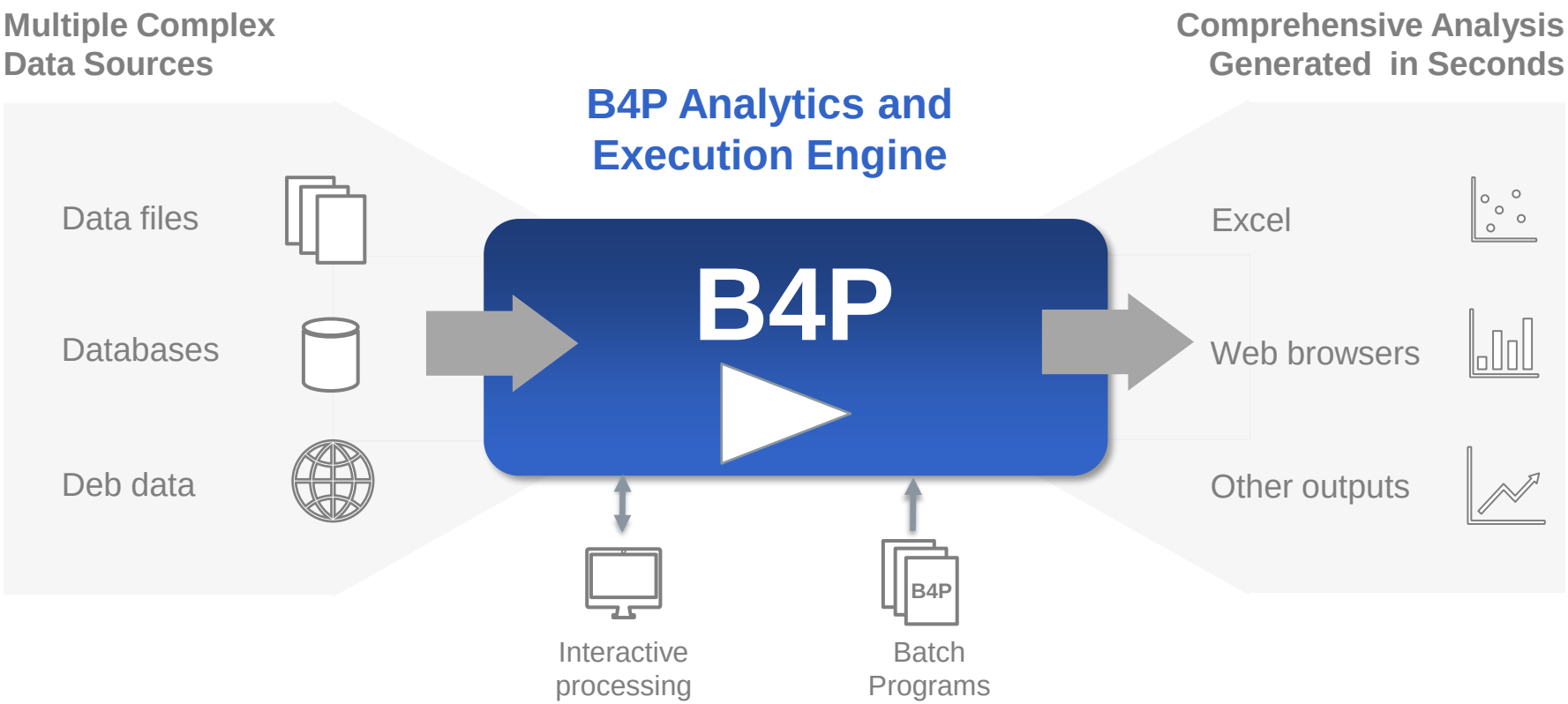
A powerful programming language and analytics engine enabling rapid transformation of big data into powerful insights

Transforming Big Data to Powerful Insights

Image source: Amir Hanna, Rush Hour in Dubai / UAE, unsplash.com.

Release 8.00 2021-01-05 "Friedrich Dürrenmatt"

B4P
The Analytics and Execution Engine – Overview



Data files: Excel, XLS/XLSX, CSV/TSV, HTML/MHTML, XML, JSON, ZIP, Text

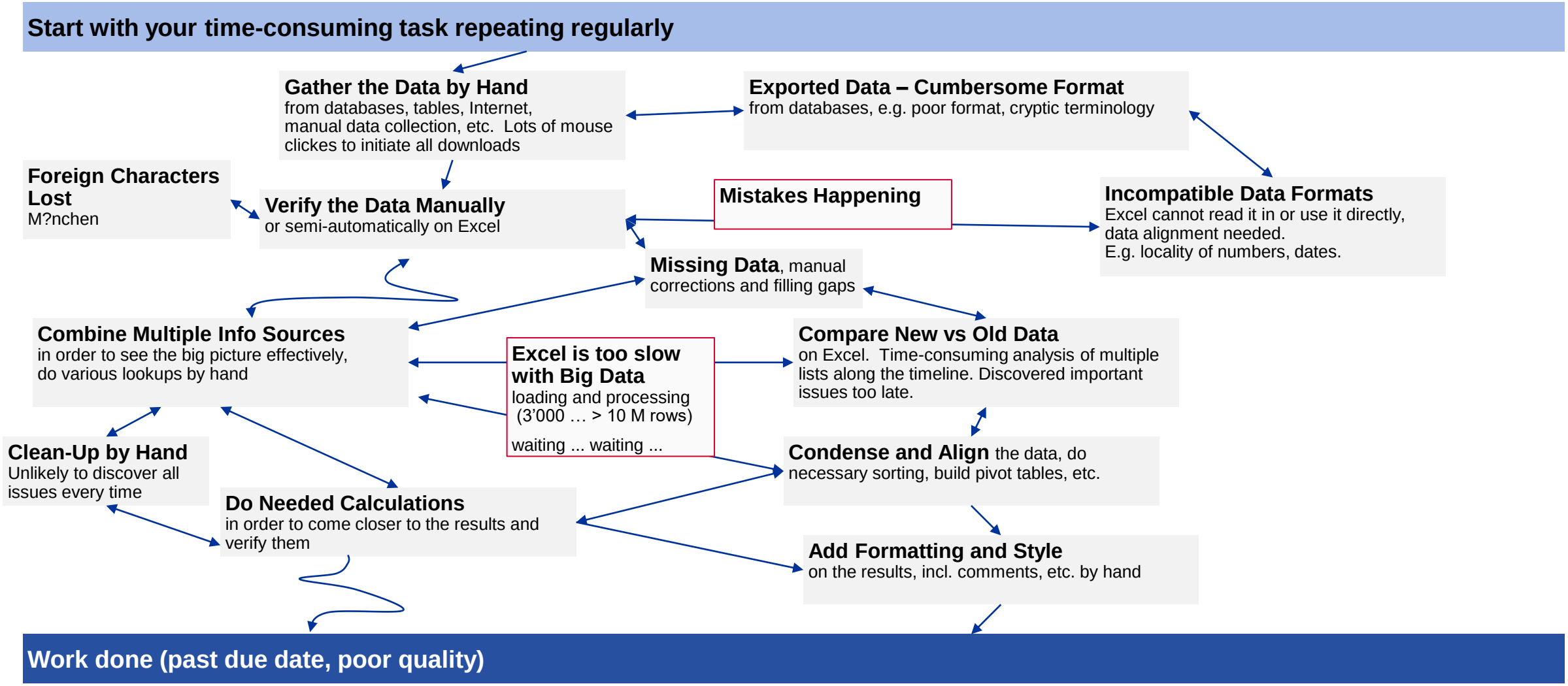
Databases: Salesforce, Oracle, SAP, Access, Filemaker, …

Web data: Data from any Internet accessible source

Table of Contents

B4P – The Problem Statement

Conventional methods do not address challenges of data management



B4P – The Problem Statement

Possible Conventional Solutions – Probably not the Best Idea

Write Excel Macros
(Visual Basic)

- OK for simple tasks, but ...
- ... coding can become cumbersome if problems are more complex. Vulnerable if data format changes.
- Processing performance will drop significantly when working with high data volumes.

Complex code and poor performance for tasks if they are not that simple

Write a Computer Program
(C, Java, Python, etc.)

- Runs very fast, but takes a lot of time to program, debug and optimize.
- Others may have difficulties to understand what you have written.
- Such programs end up very large, with many functional details coded by hand.
- Good programming know-how, ideally object-oriented programming sklls are needed, as well as obtaining a suitable development environment.

Time consuming, others having difficulties understanding your work

Hire a Consultant
(or two)

- They are happy to solve your problems for cash. Solutions are quite decent, but ...
- ... if you need further enhancements, they will ask for more cash.
- You will depend on them as they expected, and keep convincing your boss to have these expenses approved.

Time consuming, depending on others, expensive, bothering others repeatedly for approvals

B4P – The Right Approach

Automate your Work with Minimum Efforts

Start your automated tasks

Gather the raw data
- From files, databases and the Internet
- Excel
- JSON
- HTML, XML
- CSV, TSV, ...
- Text files and many more

Explore & check
- Identify and extract data you need
- Check all headers
- Identify data
- Resolve inconsistencies
- Fill information gaps
- Interpolating missing values
- Identify and resolve duplications

Clean up
- Harmonize keywords and values
- Align headers and overall structure
- Align different number formats
- Align deviating terminologies

Merge data
- Combine data from multiple tables
- Eliminate redundant data

Do some lookups
- Enrich the information value in your tables

Process and align
- Consolidate
- Analyze
- Calculate
- Pivot
- Align
- Sort

Final finish
- Sort and arrange rows and columns
- Add style, formats and colors

Save the work
- Excel
- HTML, XML
- JSON
- CSV, TSV
- Text files and many more

B4P

Analysis done in seconds

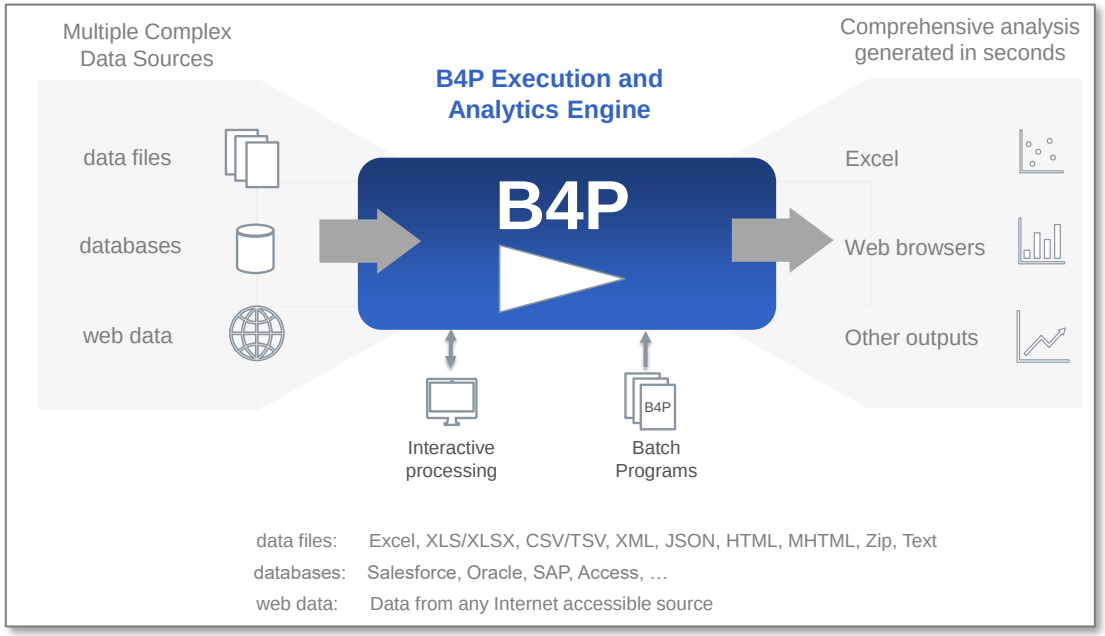
Excellent Work done! (on time, reliable results, constantly high quality)

Table of Contents

B4P – The Analytics and Execution Engine

Based on 14 Years of Experience Solving Problems

The Engine



- Lean architecture of the engine delivers **full machine performance**
- Supports **various data formats** for inputs and outputs (Excel, HTML, XML, JSON, text files, etc., full UNICODE)
- Processes and delivers **accurate results reliably**
- High performance even with big data – **Matter of seconds, not hours**
- **Styled and formatted output** for Excel and HTML (e.g. Structured tables, colors, multiple Excel sheets per file)

The Language

```
// A small demonstration program showing how Excel files (.xlsx) are read, processed

include ( Style Library );    // Include this library if you want to use the 'table

    table load excel file                    ( football club, Football Membership Li
    table load                               ( soccer club,   Soccer Membership List
    table rename column headers              ( football club, { Family Name, City },
    table process selected rows              ( soccer club,   [Level]==Novice, [Leve
    table merge                              ( football club, soccer club, { Last Na
    table sort rows                          ( soccer club,   { Level, Last Name,  F
    table rearrange columns                  ( soccer club,   { Level, First Name, L

    // Done processing.  Before saving to Excel fie, do some formatting.

    table style table                        ( soccer club, sheet, column width, 20,
                                                            freeze rows, 1, autofilt
    table style rows                         ( soccer club, 0, sheet, boldface, true
    table style columns                      ( soccer club, Level, sheet, column wid

    table process selected rows              ( soccer club, ([Level] = '*Questionabl
                table style cells                ( soccer club, Level, row(), single

    translate style attributes for excel     (soccer club);
    table save excel file                    ( soccer club, Soccer Club, New Soccer
```

Principle of Low-Code Approach: Few statements suffice

- A simple syntax: Easy to read, learn, understand and run
- Key strengths on large **structured data tables** and **hierarchically structured variables**
- Extensive library with very powerful functions and features
- Compact formulation methods for powerful processing steps minimizes coding loops and using variables

B4P – The Analytics and Execution Engine

Supported Data Formats

Inputs

Excel

- **XLSX**, **XSLM**, open formats
- **CSV** comma and tab separated files

Database Exports

- **HTML**, **MHTML** and **XML** formats (depending what the database is producing). Examples: Salesforce, Oracle, SAP
- **JSON** files (JavaScript Object Notation format)
- **CSV** comma / tab / semicolon / ...symbol separated files

Other Inputs

- Files with fixed columns on every row
- Any other form of structured text files
- **ZIP** files (B4P does data decompression)

Character Sets (both input and output)

- UNICODE UTF-8 and UTF-16; Basic and extended multilingual planes
- Legacy formats (like ASCII / Windows West Europe)

Outputs

Unformatted Output for Excel

- **CSV** comma separated files

Formatted Output for Excel (with colors, formatting and style)

- **XLSX** (Excel 2007 onwards, in use today)
- **XLS** (Excel 2003 XML format)

Unformatted and formatted output for Browsers

- **HTML** (incl. colors, formatting and style)
- **XML** (planned)

Output for other databases

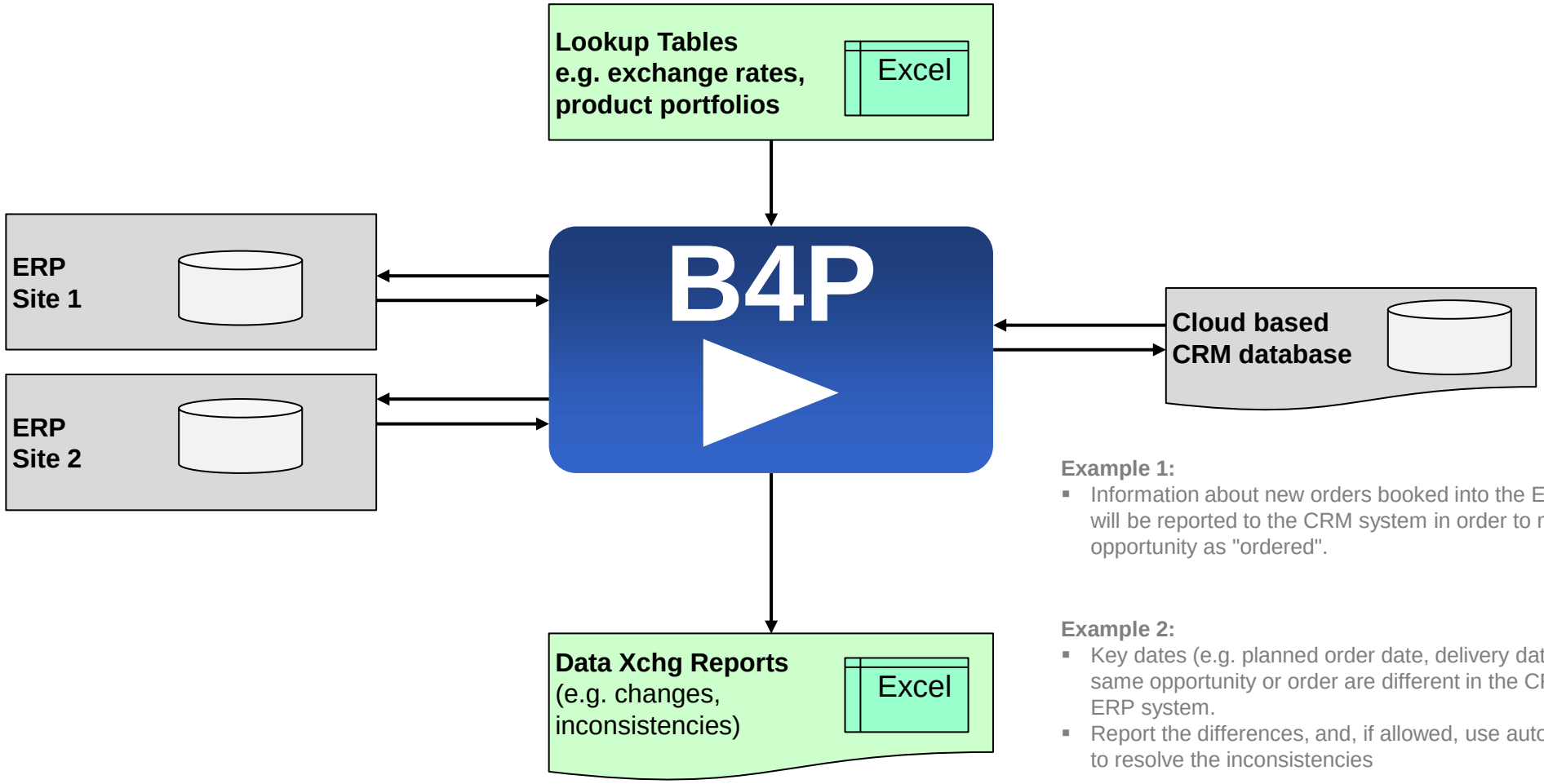
- **CSV** comma / tab / semicolon / ... symbol separated files
- **JSON** files
- Plain text files
- **ZIP** files (B4P does data compression)

Additional data formats can be supported on request, e.g. with a B4P library extension

Table of Contents

B4P – Typical Use Cases 1

Information interchange between multiple different databases



Example 1:

- Information about new orders booked into the ERP system will be reported to the CRM system in order to mark the opportunity as "ordered".

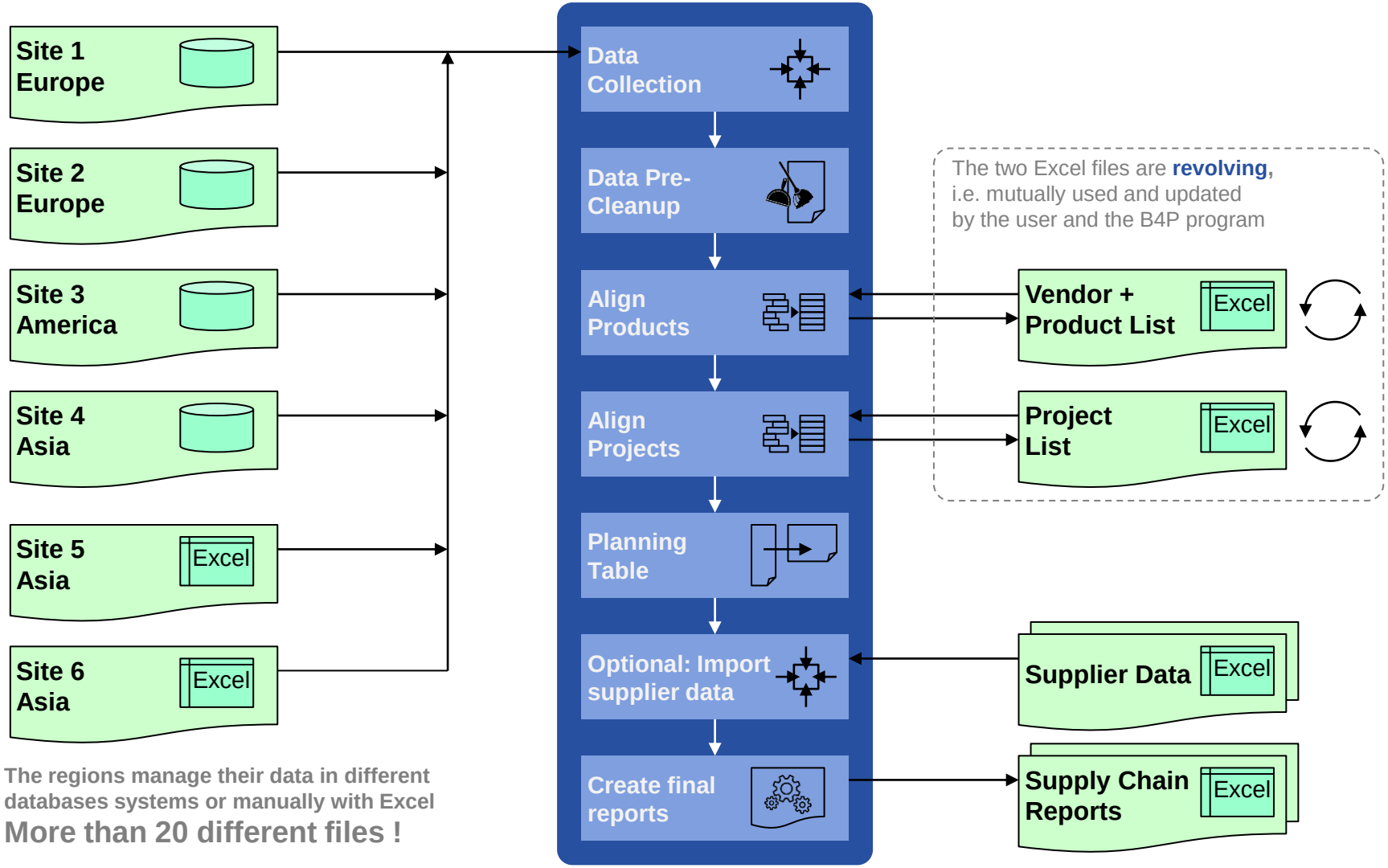
Example 2:

- Key dates (e.g. planned order date, delivery dates) for the same opportunity or order are different in the CRM and ERP system.
- Report the differences, and, if allowed, use automated rules to resolve the inconsistencies

ERP = Enterprise Resource Management Databases (e.g. SAP, Oracle, BAAN, Abacus)
CRM = Customer Relationship Database (e.g. Salesforce)

B4P – Typical Use Cases 2

Heterogeneous data integration and cleanup from different sites worldwide



The regions manage their data in different databases systems or manually with Excel

More than 20 different files !

1. Load all Demand Files
- The data from the different sites originate from different database exports or manually prepared Excel files

2. First Clean-Ups
- Harmonize data formats to week numbers and years

3. Product Alignment
- The revolving table manages the products to include and allow for using harmonized product names.
- Orientation is by common product identification number.

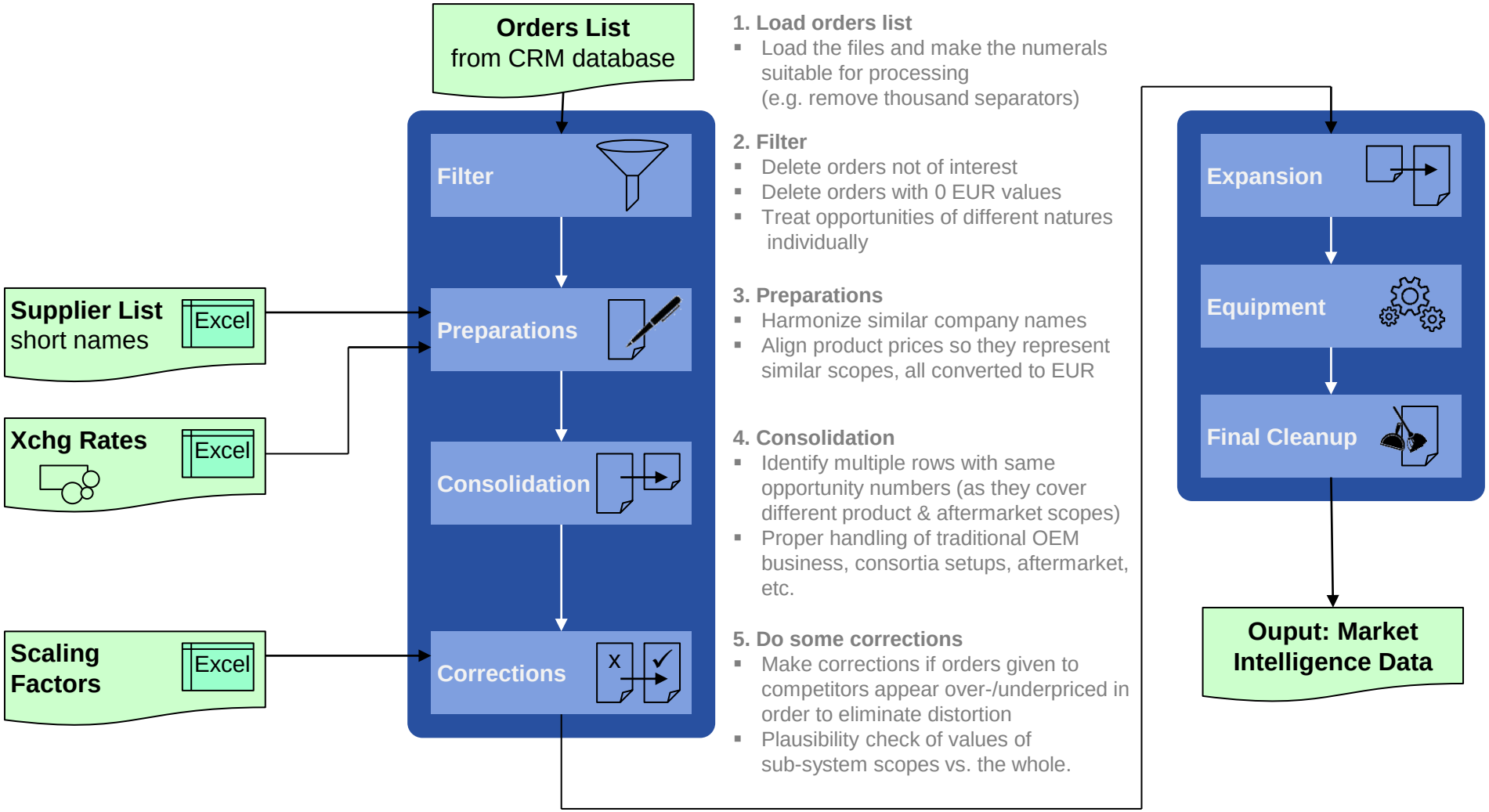
4. Project Name Alignment
- Different project names and/or abbreviations are used by the sites. They will be aligned.

5. Project Name Alignment
- The sequential list of individual demands is transformed to a horizontal planning table with weekly schedule.
- Information consolidation and summing up

6. Import supplier planning data
- If supply info is available, then match demand with their delivery plans

7. Reports
- Detailed internal reports for performance monitoring
- Condensed reports for suppliers

B4P – Typical Use Cases 3
Enriched Business Intelligence



1. Load orders list
- Load the files and make the numerals suitable for processing (e.g. remove thousand separators)

2. Filter
- Delete orders not of interest
- Delete orders with 0 EUR values
- Treat opportunities of different natures individually

3. Preparations
- Harmonize similar company names
- Align product prices so they represent similar scopes, all converted to EUR

4. Consolidation
- Identify multiple rows with same opportunity numbers (as they cover different product & aftermarket scopes)
- Proper handling of traditional OEM business, consortia setups, aftermarket, etc.

5. Do some corrections
- Make corrections if orders given to competitors appear over-/underpriced in order to eliminate distortion
- Plausibility check of values of sub-system scopes vs. the whole.

6. Expand
- Expand opportunities with multiple equipment suppliers into multiple rows and list the suppliers separately

7. Derive Equipment Values
- Derive equivalent equipment values
- Plausibility checks + adjustments against upper and lower boundaries.

8. Final cleanup
- Keep and rearrange columns of interest,
- Discard the rest
- Numeric output in commercial format

Use Excel to visualize
- Market development over time
- Market development across regions
- Market share among suppliers
- Identifying market disruptions
- etc.

ERP = Enterprise Resource Management Databases (e.g. SAP, Oracle, BAAN, Abacus)
CRM = Customer Relationship Database (e.g. Salesforce)

Table of Contents

B4P – The Language
Key Benefits of a Low-Code Language Concept

A language allowing you to express yourself briefly to solve the problem. Focus on the *what*, not *how*.

- **Simple** procedural language
- **Very easy to read and understand** the code, therefore very easy to learn programming.
- Powerful language semantics **keeps your program short to solve the problem**.
- **Very quickly** to get your code running
- **Be natural** and less cryptic. Giver your variables, tables, functions, etc. natural names (spaces are allowed !)
- **Tables** of any size are one of the main data storage models and B4P is optimized for this.
- **No programming bureaucracy** such as type definitions, declaring all the variables and doing memory management on your own.
- Significantly less need for fine grained programming like formulating loops, using variables, coding detailed algorithms, etc.
- B4P understands data formats such as **Excel, HTML, XML, JSON, CSV**, etc. to retrieve data from Excel, database and the Internet directly
- The execution engine and all library files are **very light-weight and lean**, very robust and start quickly.
- High focus on **cross-platform portability** (Windows, Linux, MacOS, etc.), enabling to run the same code on different platforms.
- B4P output files for **Excel with style and formatting** like colors, row widths, etc.

- **Rich B4P function library** with ca 800 functions, including 200 functions for processing tables, and growing.

B4P – The Language
Language Syntax and Semantics

- Overall language block structure similar to C / C++ / Java.
- Full and homogeneous UNICODE support.
- Tables and structured variables are the two main data storage mechanisms.
- Table names, variable names and function names are fully flexible, e.g. multiple words and spaces are allowed.
  - This allows you to load sophisticated JSON data tree structures into the hierarchical variable structure.
- Dynamic variable tree, allowing to build up nested arrays and structures.
- Code pieces can be passed as function parameters which will be executed multiple time or on a on-demand basis:
  Example: table process (...), pick if (...)
  Benefit: Eliminates need to write loops or other details
- Numerous flexible control flow mechanisms, going beyond the common ones like if, while, for, ...
- File names with directory paths are understood and interpreted correctly in other platforms (e.g. Windows vs. Linux).
- Powerful parameter set and matrix operations to process big data.

When you program, think big:

Hint 1 **Use the rich B4P function library** to process these tables and big data.
Large tables will be analyzed and processed at naked machine performance.

Hint 2 **Use deep operations** (vector and matrix operations) to process large amount of data inside tables and parameter sets.

Hint 3 Think how you can formulate your code in a very compact manner without compromising comprehensibility.
Doing the great things with 5 - 20 lines of code is within reach.

Table of Contents

B4P – Program Example 1

Merging 2 Tables – Problem Statement

Football Membership List.xlsx

First Name	Family Name	City	Level
Abel	Amberstone	Amsterdam	Beginner
Beata	Berghill	Barcelona	Experienced
Corinne	Carlson	Copenhagen	Beginner
Dietmar	Davis	Dublin	Beginner
Ellen	Evans	Essen	Beginner
Fred	Fisher	Frankfurt	Experienced
Gregory	Green	Gaza City	Experienced
Henry	Hansson	Hamburg	Experienced
Ida	Ingelberg	Ingolstadt	Beginner
John	Janssen	Johannesburg	Beginner
Karl	Karlsson	Kansas City	Experienced

Soccer Membership List.xlsx

Level	Town	Last Name	First Name
Questionable	Kyoto	Karlsson	Karl
Novice	London	Lee	Linda
Experienced	Morristown	Miller	Mike
Experienced	New York	Nguyen	Nathali
Experienced	Oslo	Oliveiro	Oscar
Novice	Phoenix	Paulsson	Petra
Novice	Quebec City	Quarles	Quincy
Experienced	Riga	Richardson	Richard
Experienced	San Diego	Stewart	Sandra
Experienced	Tahoma	Turner	Tim
Questionable	Ulm	Ufford	Uwe
Novice	Venice	Viking	Victor

New Soccer Membership List.xlsx

Level	First Name	Last Name	Town
Beginner	Abel	Amberstone	Amsterdam
Beginner	Corinne	Carlson	Copenhagen
Beginner	Dietmar	Davis	Dublin
Beginner	Ellen	Evans	Essen
Beginner	Ida	Ingelberg	Ingolstadt
Beginner	John	Janssen	Johannesburg
Beginner	Linda	Lee	London
Beginner	Petra	Paulsson	Phoenix
Beginner	Quincy	Quarles	Quebec City
Beginner	Victor	Viking	Venice
Experienced	Beata	Berghill	Barcelona
Experienced	Fred	Fisher	Frankfurt
Experienced	Gregory	Green	Gaza City
Experienced	Henry	Hansson	Hamburg
Experienced	Mike	Miller	Morristown
Experienced	Nathali	Nguyen	New York
Experienced	Oscar	Oliveiro	Oslo
Experienced	Richard	Richardson	Riga
Experienced	Sandra	Stewart	San Diego
Experienced	Tim	Turner	Tahoma
Questionable	Uwe	Ufford	Ulm
Questionable or Experienced	Karl	Karlsson	Kyoto or Kansas City

A new football club should be created after merging two existing sports clubs:

- The tables of the two clubs are arranged differently and contain different acronyms (e.g. qualification levels)
- Some people are members in both clubs and need to be resolved properly.
- Highlight possible inconsistencies (red text color)

B4P – Program Example 1

A Simple Program Merges 2 Tables – 15 Statements, 0 Variables, 0 Loops

Simple statements easy to understand

No loops. Simple statements applied for the whole table

Excel file loaded with one single statement

Multiple words allowed for readability + flexibility

1 statement merges 2 tables as specified

7 Statements to combine the clubs

7 statements to add some formatting and save the work.

```
// A small demonstration program showing how Excel files (.xlsx) are read, processed and saved, including some style.

include ( Style Library );    // Include this library if you want to use the 'table style ...' functions.

    table load excel file                    ( football club, Football Membership List.xlsx );
    table load                               ( soccer club,   Soccer Membership List.csv ); // Beginners are Novices here
    table rename column headers              ( football club, { Family Name, City }, { Last Name, Town } );
    table process selected rows              ( soccer club,   [Level]==Novice, [Level]=Beginner );
    table merge                              ( football club, soccer club, { Last Name, First Name }, { Level, Town }, append, " or " );
    table sort rows                          ( soccer club,   { Level, Last Name,  First Name });
    table rearrange columns                  ( soccer club,   { Level, First Name, Last Name, Town } );

    // Done processing.  Before saving to Excel fie, do some formatting.

    table style table                        ( soccer club, sheet, column width, 20, row height, 20,
                                                            freeze rows, 1, autofilter, 0, vertical align, center );
    table style rows                         ( soccer club, 0, sheet, boldface, true, fill color, gray 15 );
    table style columns                      ( soccer club, Level, sheet, column width, 30 );

    table process selected rows              ( soccer club, ([Level] = '*Questionable*'),
                table style cells                ( soccer club, Level, row(), single, text color, red ) );

    translate style attributes for excel     (soccer club);
    table save excel file                    ( soccer club, Soccer Club, New Soccer Club Membership List.xlsx );

    echo ("New soccer club has ", table length( soccer club ), " members. Enjoy playing.");
```

Excel file saved with one single statement

B4P – Program Example 2
U.S. Presidents from Wikipedia

destroy the fragile unity holding the nation together, Washington remained unaffiliated with any political faction or party throughout his eight-year preside
political party.[2]

Contents [hide]

Presidents

	Presidency[a]	President		Party[b]	Election	Vice President
1	April 30, 1789 – March 4, 1797		George Washington	Unaffiliated	1788–89, 1792	John Adams[c]
2	March 4, 1797 – March 4, 1801		John Adams	Federalist	1796	Thomas Jefferson[d]
					1800	Aaron Burr

B4P

Presidenc	Presidency (1)	President	Party	Election	Vice President
1	April 30, 1789 – March 4, 1797	George Washington	Unaffiliated	1788–89, 1792	John Adams,
2	March 4, 1797 – March 4, 1801	John Adams	Federalist	1796	Thomas Jefferson
3	March 4, 1801 – March 4, 1809	Thomas Jefferson	Democratic-Republican	1800, 1804	Aaron Burr, George Clinton
4	March 4, 1809 – March 4, 1817	James Madison	Democratic-Republican	1808, 1812	, Vacant after Apr. 20, 1812, Elbridge Gerry, Vacant after Nov. 23, 1814
5	March 4, 1817 – March 4, 1825	James Monroe	Democratic-Republican	1816, 1820	Daniel D. Tompkins,
6	March 4, 1825 – March 4, 1829	John Quincy Adams	Democratic-Republican	1824,	John C. Calhoun,
7	March 4, 1829 – March 4, 1837	Andrew Jackson	Democratic	1828, 1832	, Vacant after Dec. 28, 1832, Martin Van Buren
8	March 4, 1837 – March 4, 1841	Martin Van Buren	Democratic	1836	Richard Mentor Johnson
9	March 4, 1841 – April 4, 1841	William Henry Harrison	Whig	1840	John Tyler
10	April 4, 1841 – March 4, 1845	John Tyler	Whig		Vacant throughout presidency,
11	March 4, 1845 – March 4, 1849	James K. Polk	Democratic	1844	George M. Dallas

. . .

44	January 20, 2009 – January 20, 2017	Barack Obama	Democratic	2008, 2012	Joe Biden,
45	January 20, 2017 – Incumbent	Donald Trump	Republican	2016	Mike Pence

Download the list of Presidents and generate Excel table with 1 President per row

- Some Presidents had multiple terms
- Ignore the portraits
- Some vice presidents had deviating terms
- Remove redundant artefacts, e.g. cross-referencing symbols and generate a nice table with **parties colored differently**

Web source: https://en.wikipedia.org/wiki/List_of_presidents_of_the_United_States

B4P – Program Example 2

18 Statements, 0 Loops and 0 Variables to Straighten up the Presidents

Internet Download in 1 statement

Cleanups with minimum effort

Align the data

Define colors affiliated to the parties

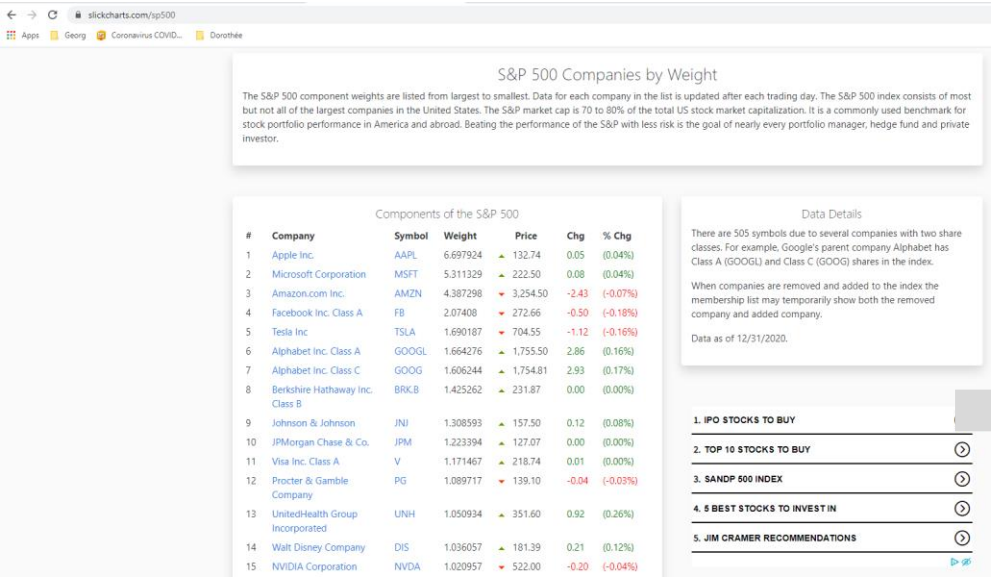
Color and Style the results

Save as Excel File

```
include ( Style Library ); // This library needs to be included if you want to add style and formatting to the table

      file download overwrite       ( "https://en.wikipedia.org/wiki/List_of_presidents_of_the_United_States", presidents.html );
      table load                    ( presidents, presidents.html, HTML, 'id="Presidents"' );
      table save                    ( presidents, presidents as loaded.csv );
      table delete rows             ( presidents, table length( presidents ) -1 );  // Remove last row with redundant footnote info

      // Strip out all footnote references and new lines in the fields
      table process all cells       ( presidents, [.] = replace all( literal([.]), { "[a]" .. "[z]", new line}, '' ) );

      // Remove the blank column originally containing portraits and put president name into all rows

      table delete columns          ( presidents, {President, Party } );
      table rename column headers   ( presidents, {"President (1)","Party (1)"}, {President, Party} );
      table fill vertically         ( presidents, President );
      table consolidate             ( presidents, President, { Election, Vice President }, append, ", " );

      table initialize ( party colors,    // Define party colors
            {{ Party Name,                Colors },
             { Democratic,                "#8080FF" }, // Light blue
             { Republican,                "#FF8080" }, // Light red
             { Federalist,                coral },
             { "Democratic- Republican",  excel light green },
             { "Democratic-Republican",   excel light green },
             { Whig,                      yellow },
             { National Union,            ocre },
             { Unaffiliated,              gray 15 } } );

      // Add some colors and styles
      table process                 ( presidents, table style cells( presidents, Party, row(), single,
                                                  fill color, [party colors : Party Name, [Party], Colors ] ) );
      table style columns           ( presidents, { "Presidency (1)", "President", "Vice President" }, sheet, column width, 30 );
      table style columns           ( presidents, { Party, Election }, sheet, column width, 20, horizontal align, middle );
      table style rows              ( presidents, 0, table, boldface, true );
      table style table             ( presidents, sheet, wrap text, true, autofilter, 0, freeze rows, 1 );


      translate style attributes for excel      ( presidents );
      table save excel file                     ( presidents, All U.S. Presidents, presidents.xlsx );
```

B4P – Program Example 3

Stock Data (SP 500 and NASDAQ 100) Combined



B4P

#	Company	Symbol	Weight	Price	Chg	% Chg
1	Apple Inc.	AAPL	6.697924	132.74	0.05	(0.04%)
2	Microsoft Corporation	MSFT	5.311329	222.50	0.08	(0.04%)
3	Amazon.com Inc.	AMZN	4.387298	3,254.50	-2.43	(-0.07%)
4	Facebook Inc. Class A	FB	2.07408	272.66	-0.50	(-0.18%)
5	Tesla Inc	TSLA	1.690187	704.55	-1.12	(-0.16%)
6	Alphabet Inc. Class A	GOOGL	1.664276	1,755.50	2.86	(0.16%)
7	Alphabet Inc. Class C	GOOG	1.606244	1,754.81	2.93	(0.17%)
8	Berkshire Hathaway Inc. Class B	BRK.B	1.425262	231.87	0.00	(0.00%)
9	Johnson & Johnson	JNJ	1.308593	157.50	0.12	(0.08%)
10	JPMorgan Chase & Co.	JPM	1.223394	127.07	0.00	(0.00%)
11	Visa Inc. Class A	V	1.171467	218.74	0.01	(0.00%)
12	Procter & Gamble Company	PG	1.089717	139.10	-0.04	(-0.03%)
13	UnitedHealth Group Incorporated	UNH	1.050934	351.60	0.92	(0.26%)
14	Walt Disney Company	DIS	1.036057	181.39	0.21	(0.12%)
15	NVIDIA Corporation	NVDA	1.020957	522.00	-0.20	(-0.04%)

#	Company	Symbol	Price	Chg	% Chg	Weight sp500	Weight nasdaq10(
1	3M Company	MMM	175	0.21	0.12%	0.318459	
2	A. O. Smith Corporation	AOS	54.82	0	0.00%	0.023472	
3	ABIOMED Inc.	ABMD	326.6	2.4	0.74%	0.046276	
4	AES Corporation	AES	23.7	0.2	0.85%	0.049417	
5	AMETEK Inc.	AME	120.94	0	0.00%	0.087884	
6	ANSYS Inc.	ANSS	363.8	0	0.00%	0.098729	0.258
7	ASML Holding NV	ASML	486.99	-0.73	-0.15%		0.305
8	AT&T Inc.	T	28.76	0	0.00%	0.647313	
9	AbbVie Inc.	ABBV	107.15	0	0.00%	0.597494	
10	Abbott Laboratories	ABT	109.49	0	0.00%	0.612924	
11	Accenture Plc Class A	ACN	261.21	0	0.00%	0.522713	
12	Activision Blizzard Inc.	ATVI	92.85	0	0.00%	0.226706	0.593
13	Adobe Inc.	ADBE	500.1	-0.02	0.00%	0.757774	1.984
14	Advance Auto Parts Inc.	AAP	157.51	0	0.00%	0.033788	
15	Advanced Micro Devices Inc.	AMD	91.66	-0.05	-0.05%	0.348385	0.912
16	Aflac Incorporated	AFL	44.47	0	0.00%	0.091757	
17	Agilent Technologies Inc.	A	118.49	0	0.00%	0.114563	
18	Air Products and Chemicals Inc.	APD	273.22	0	0.00%	0.190791	
...							
526	Zoom Video Communications Inc	ZM	339.2	1.88	0.56%		0.543
527	eBay Inc.	EBAY	50.11	-0.14	-0.28%	0.103937	0.286
528	salesforce.com inc.	CRM	222.53	0	0.00%	0.643119	

Import the SP 500 and NASDAQ 100 listings and merge them

- Some companies are listed in only one of them, others are listed in both.
- Combine the information, show the weighting in the two listings and color the stock price developments

Web source: https://www.slickcharts.com/nasdaq100, https://www.slickcharts.com/sp500

B4P – Program Example 3

21 Statements, 1 loop and 1 variable do the Job

Download Files
and align header names

Merge the Tables
and assign common name

Clean Up

Add style and color

Save as Excel File

```
include ( Style Library );

//    Step 1: Download two web pages.
      for all parameters ( {nasdaq100, sp500} , listing[] )
      {
            file download overwrite       ( "https://www.slickcharts.com/" + listing[], listing[] + .html);
            table load                    ( listing[] ,listing[] + .html,  HTML, "Components of the" );
            table rename column headers   ( listing[], "Weight", "Weight " + listing[] );
                                          // Weight info is specific to SP and Nasdaq, so add the listing name
            table save                    ( listing[], listing[] + " as loaded.csv" );
      }

//    Step 2:     Combine the two tables
      table merge extend columns    ( nasdaq100, sp500, Symbol );
      table rename                  ( sp500, stocks );

//    Step 3: Cleanup
      table correct headers         ( stocks, '*Price*', Price );
      table rearrange columns       ( stocks, { '#', Company, Symbol, Price, Chg, '% Chg' } ); // Weightings follow afterwards

// Percent value gets converted to regular number, price value is claned up
      table process           ( stocks, ['% Chg'] = smart numeral( middle( ['% Chg'], '(', ')' )); [Price] = clean numeral([Price]) );
      table sort rows         ( stocks, Company );
      table process           ( stocks, ['#'] = row() );

//    Step 4: Add some color
      table process           ( stocks,
          table style cells     ( stocks, {'Chg','% Chg'}, {2:row()}, single, text color, select if ( [Chg]>0, excel green, red ) ) );

      table style rows        ( stocks, 0, table, fill color, gray 14, boldface, true, wrap text, true );
      table style columns     ( stocks, Company, sheet, column width, 30 );
      table style columns     ( stocks, '% Chg', sheet, number format, "0.00%" ); // Value to show as percent.
      table style table       ( stocks, sheet, freeze rows, 1, autofilter, 0);

//    Step 5: Save the artwork
      translate style attributes for excel      ( stocks );
      table save excel file                     ( stocks, "NASDAQ and SP500", Stocks.xlsx );
```

B4P

Beyond Former Performance.

Information *www.b4p.app*

Contact

Europe
Georg zur Bonsen
+41 56 221 82 00
zur-bonsen (at) bluewin .ch

North America
Rafael Richards
+1 202 469 15 27
rmrich5 (at) gmail .com

Presentation: Updated 2021-01-07

Table of Contents

B4P – Beyond Former Performance

New in Release 8.00

100% Excel Support

- B4P loads and saves the latest Microsoft Excel file format (.xlsx)
- Formatting and styles are supported
- B4P also supporting the old Excel XML 2003 file format
- Round trip: Generated excel files can be loaded again with out losing data.

Excel files are actually ZIP files containing various files describing workbooks, tables, styles and shared strings.

The functions
- **table load excel file** and
- **table save excel file**

are implemented in B4P.

Overall Clean-up

- Thorough functional tests and stress tests have been carried out to make B4P become even more reliable.
- The start-up behavior has been streamlined.

A small number of new functions have been added which include the following:

- Processing table columns
- Processing table cells
- Advance table search and update functions
- Improved visualization of tables
- Improved listing of tables, functions and variables

Help and Documentation

- Moved from PDF to 100% online
- www.b4p.app
- Online help available inside B4P on your fingertips, i.e. by typing "docs"
- Lots of reproducible program examples
- Various new help features are available

The document generation has been automated using a B4P program. It fishes all raw contents from source code comments and supplementary text files, generates a master file (JSON) and finally produces all web document contents automatically.

Almost all programming examples are tested automatically with every new update and the outputs are included in the documentation.

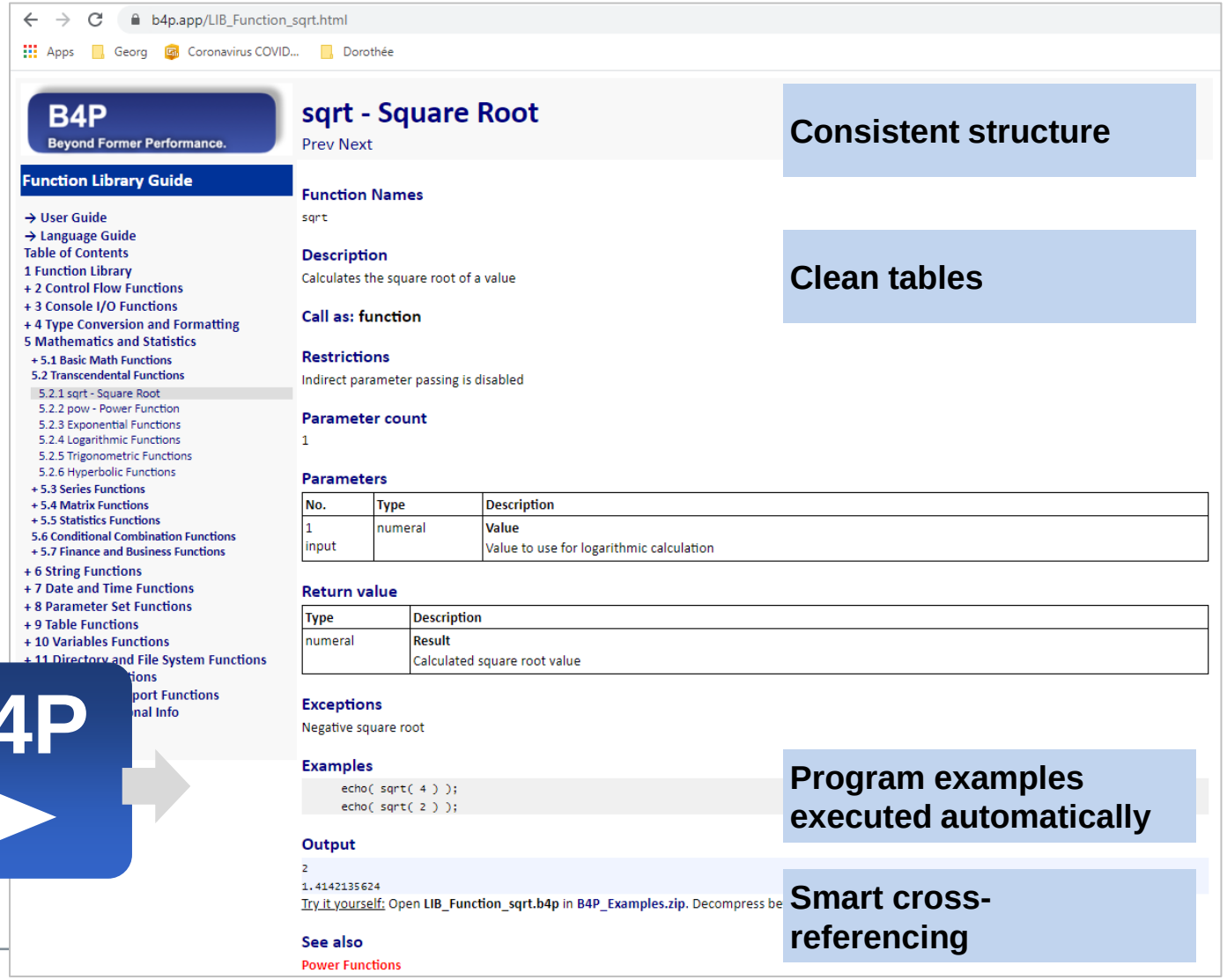
B4P Use Case

Automatic documentation generation for website *www.b4p.app*

Inputs: Source Text (inside C program)

```
//    sqrt
//  ____________________________________________________________________

/* B4PDOCU.START

  "sqrt - Square Root" :
  {
      "Function Names": "sqrt",
      "Keywords":       "Square Root",
      "Documentation":  "Function Description",
      "Description"::
            Calculates the square root of a value
      +++,
      "Call as":                "function",
      "Parameter count":        "1",
      "Restrictions":           "Indirect parameter passing is disabled",
      "Parameters":
      [
            { "Number":        "1",
              "Name":          "Value",
              "Direction":     "input",
              "Types":         "numeral",
              "Description":   "Value to use for logarithmic calculation"
            } ],
      "Return value":
      [
            { "Name":          "Result",
              "Types":         "numeral",
              "Description":   "Calculated square root value"
            }
      ],
      "Exceptions": "Negative square root",
      "Examples"::
            echo( sqrt( 4 ) );
            echo( sqrt( 2 ) );
      +++,
      "Output":   "automatic",
      "See also": "Power Functions"
}

B4PDOCU.STOP */

void function__sqrt(int bias, Token_List& p, Token_Entry& ret_val)
{
      ...
      return;
}
```

Output: Web contents



Consistent structure

Clean tables

Program examples executed automatically

Smart cross-referencing

B4P Use Case

Automatic Document Generation for www.b4p.app using B4P

Catalog of Books

- Short Excel File
- Specifies all manuals to generate
- Specifies location of source text for the different manuals

Raw Inputs

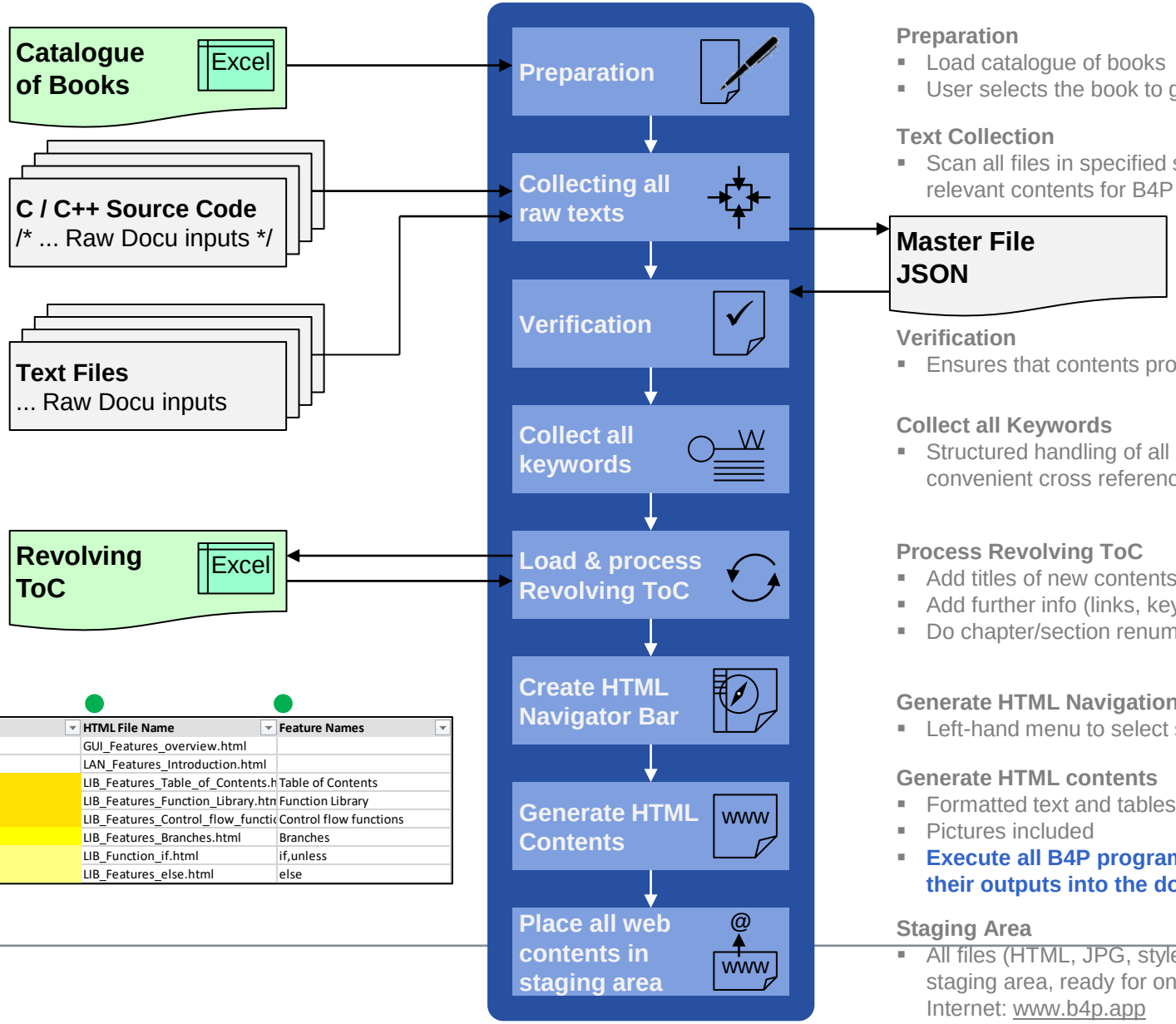
- Descriptions of B4P functions are documented in the C/C++ files in comments, using an enhanced JSON format.
- Other contents such as introductory parts are described in additional text files, also using enhanced JSON format

Revolving Table of Contents

- It puts all individual raw document sections into a given order and hierarchy level in the document
- Both user ● and B4P program ● update this table mutually

● ● ● ● ●●

Status	Category	Level	Section N	Section Name	HTML File Name	Feature Names
Missing, (Ass	Link			→ User Guide	GUI_Features_overview.html	
Missing, (Ass	Link			→ Language Guide	LAN_Features_Introduction.html	
OK, (Assign r	TOC	1		Table of Contents	LIB_Features_Table_of_Contents.h	Table of Contents
OK	Body	1	B 1	Function Library	LIB_Features_Function_Library.htm	Function Library
OK	Body	1	B 2	Control Flow Functions	LIB_Features_Control_flow_functi	Control flow functions
OK	Body	2	B 2.1	Branches	LIB_Features_Branches.html	Branches
OK	Body	3	B 2.1.1	if, unless	LIB_Function_if.html	if,unless
OK	Body	3	B 2.1.2	else	LIB_Features_else.html	else



Preparation

- Load catalogue of books
- User selects the book to generate

Text Collection

- Scan all files in specified subdirectories for relevant contents for B4P documentation

Verification

- Ensures that contents provided fulfill the structural guidelines

Collect all Keywords

- Structured handling of all keywords and function names allowing convenient cross referencing and index pages

Process Revolving ToC

- Add titles of new contents into placeholders or at the bottom
- Add further info (links, keywords)
- Do chapter/section renumbering

Generate HTML Navigation Bar

- Left-hand menu to select section to see

Generate HTML contents

- Formatted text and tables
- Pictures included
- **Execute all B4P program examples automatically and add their outputs into the doc contents**

Staging Area

- All files (HTML, JPG, style.css, PDF, etc.) are moved to the staging area, ready for one-mouseclick publication on the Internet: www.b4p.app