



*GA*



10/18/2017

# MEISTER SMART™ BUSINESS APPLICATIONS FOR SAP WHITE PAPERS: REPORTING

Product Document v1.0



*GA*

Andre Rosenthal

[www.s4meister.com](http://www.s4meister.com)



# **Meister Smart™ Business Applications for SAP White Papers: Reporting**

Gateway Architects Product Document

Document Name:	Meister Suite White Papers - Disruption at SAP
Document ID:	WHPSR0003
Document Owner:	Andre Rosenthal
Document Version:	V1.0
Document Date:	10/18/2017
Document Status:	Released



**Meister Smart™ Business Applications for SAP White Papers:  
Reporting**

**Table of Contents**

**Document Control**..... 4  
    Deliverable Information ..... 4  
**Overview** ..... 5  
**The enablement of ad hoc reporting** ..... 5  
**Consumption Process**..... 6



**Meister Smart™ Business Applications for SAP White Papers:  
Reporting**

**Document Control**

**Deliverable Information**

<b>Deliverable Name</b>	<b>Meister Suite White Papers – Reporting</b>
<b>Author</b>	<b>Gateway Architects</b>
<b>Status</b>	V1.0
<b>Location</b>	

**Revision History**

<b>Version No.</b>	<b>Date</b>	<b>Author</b>	<b>Revision Description</b>
V1.0	10/18/2017	ARosenthal	Initial version

**Authorisers for sign off**

<b>Role</b>	<b>Name</b>	<b>Date Signed Off in SharePoint</b>
CTO	Andre Rosenthal	10/18/2017

**Distribution - in addition to Authorisers**

<b>Name</b>	<b>Role</b>
Yaron Meefeld	CEO
William Mays	Chairman
Craig Twiss	President, US

**Reference Documents**

<b>Document</b>	<b>Location</b>



# Meister Smart™ Business Applications for SAP White Papers: Reporting

## Overview

SAP's Reporting is enabled via execution of batch programs that could either run in foreground or background processors, and notoriously consume a large amount of NetWeaver resources. As far as enabling Reporting to be executed outside the SAP GUI, the SAPUI5 environment allows the developer to enable single instances of a given report by means of promoting the associated transaction code as a UI5 web part as part of the solution. However, there is no native support for ad hoc report selections and reports not associated with a specific transaction code, which are ran via transaction code SE38 or scheduled via transaction code SM36 and SM37.

Meister introduces another disruptive technology with its ad hoc report execution engine, which is the focus of this white paper.

## The enablement of ad hoc reporting

Meister allows the user to choose any report or associated transaction code to be executed with minimal efforts by means of enabling the creation of the exact same report selections as UI5 components (although the same functionality could be achieved in any UX environment of choice.)

Let the set of fields required to execute a report be the set described below:

$$S(F) = \sum p(a, b, \dots, i)$$

**Equation 1**

*where  $p(a, b, c \dots, i)$  is a polynomial of choices  $c$  such that  $c = g(j)$  represents what a field  $j$  represents for such report*

The function  $c = g(j)$  is nothing else but a tuple representing possible values for the given choice which could be a constant whose UX element is a simple text box, a date, a range of values found in a given table, as so forth.

The polynomial  $p(a, b, c \dots, i)$  is therefore the combined set of fields for the report which could be directives for the report execution (such as layouts), or field choices as defined by the function above.

If such polynomial could be constructed, it becomes straightforward that equation 1 is a Json Document explaining how to reconstruct the original UI in Dynpro present at SAP now transposed as UX elements using the framework chosen by the UX developer.



## **Meister Smart™ Business Applications for SAP White Papers: Reporting**

With the report now transposed into UX elements, the user could fill in the values with the support of the associated helps for the fields, and let the set below represent the values of the fields chosen at runtime by a user:

$$S(F) = \sum P(a_j, b_j, c_j, \dots i_j)$$

**Equation 2**

Using the same logic, it is clear that equation 2 is nothing else but equation 1 where specific values are sent back to Meister using the same underlying structure as instructed by equation 1.

Possessing this information now, Meister schedules in the background the execution of the report and monitors the execution with statuses enabling the user to know when a given report has finished and is ready for consumption.

### **Consumption Process**

Meister returns to the UX the report in 3 different formats:

- 1) The XML generated by SAP
- 2) The raw text in text format
- 3) The equivalent Json Document representing the report.

The 3 modes of reporting allow the further consumption of the report by any analytical dashboard capable of consuming Json Documents for graphical depiction, such as Tableau and Power BI. The raw text is sent back as a means to achieve deep learning content by means of machine learning tools such as R or TensorFlow, and the XML representation is present to enable the UX to natively show the report as it was generated by SAP.

It is worthwhile to mention that Meister is the only tool that possess the knowledge to generate reports based on existing Variants as well as ad hoc representations of any existing report outside the realm of SAP and not hardcoded by a developer.