FUNCTIONAL DEFINITION DOCUMENT

NETRONIC HTML5

Visual Scheduling Widget - Standard Edition

(VSW SE)

Version: **3.0** | State of December 1st, 2019

NETRONIC Software GmbH
Pascalstraße 15
52076 Aachen
Germany

Tel: +49 (2408) 141 0
Fax: +49 (2408) 141 33
Web: www.netronic.com

Table of contents

1 Scope of Document

The purpose of this document is to outline the functional definition of the NETRONIC HTML5 Visual Scheduling Widget, Standard Edition that NETRONIC has developed. This widget is based on HTML5/JavaScript-Technology and will be called VSW SE for short.

The Functional Definition Document (FDD) is complemented by the Interface Definition Document (IDD), in which NETRONIC will specify the API of the VSW SE. The IDD will specify all objects, events and functions that a developer can use to integrate and work with the VSW SE.

2 Purpose of VSW SE

NETRONIC provides the VSW SE as a jQuery UI widget. It is explicitly mentioned that this is not a ready-to-use application like the standard Visual Production Scheduler add-in developed by NETRONIC for standard NAV / Business Central.

3 Terminology

In this document the terminology of NETRONIC's data model is used. Following the core terms of this data model are defined:

- **Activity**: Activities cover all actions that need to be accomplished within a certain period of time to achieve a specific objective.
- **Resource**: Resources – such as machines or personnel – are the ones that are needed to perform the activities. Besides its general availability defined by a calendar, a resource has a capacity and a load.
- **Allocation**: Allocations define the assignments of resources to activities and vice versa.
- **Allocation entry**: if an activity consists of a setup time, a pure runtime, and a teardown time, then a resource may only be required for setup and teardown, but not at runtime. For this purpose, the developer has to split allocations and additionally can use more than one allocation entry. Each entry has its own start and end date and represents a time span.

4 Technology & Architecture

4.1 Technology

The VSW SE is implemented as jQuery UI Widget using the NETRONIC Web Application Framework (NWAF), which enables NETRONIC to effectively develop visual scheduling applications based on HTML5/JavaScript technology. This technological foundation has two implications:

- The usage of HTML5/JavaScript technology allows using the VSW SE in (theoretically any) device that has a web browser installed. As a Gantt chart by its nature it always requires a comparably large screen with high resolution. It is not recommend to use the VSW SE with a smartphone client despite the general technological feasibility to do so.

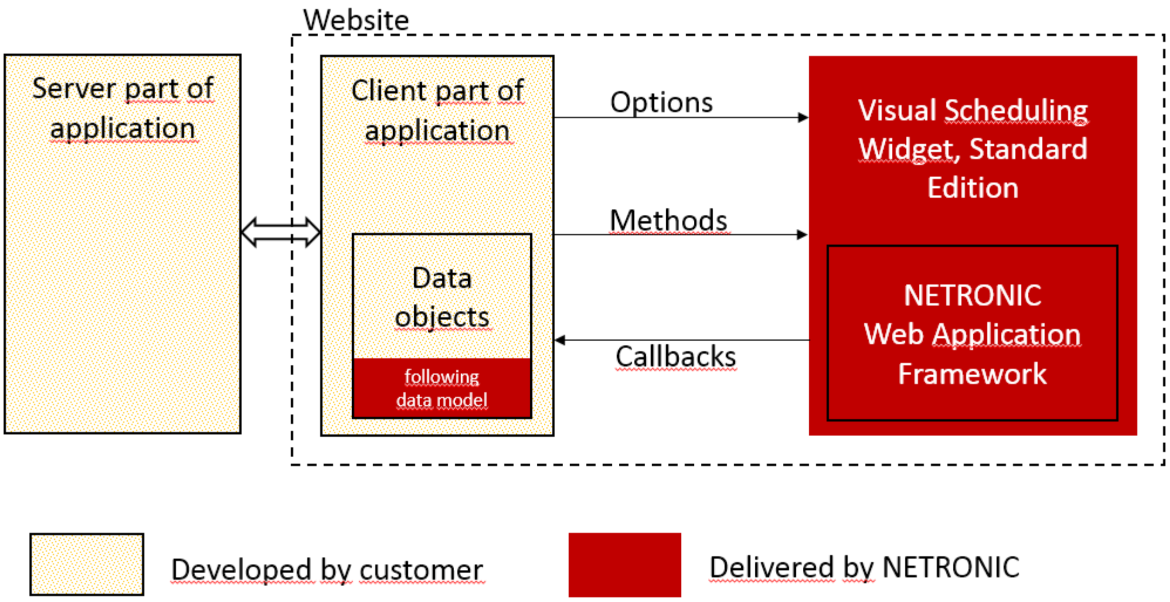
- The technology used by NETRONIC is “touch enabled”, but not “touch optimized”. This means that the VSW SE can get used on tablet PCs. However, not each native touch gestures will get supported.

NETRONIC will not deliver a stand-alone application. The VSW SE will come with an API, which can be called by an interface. This interface can be used to provide data to the VSW SE and to get callbacks from it. Handling these data within the VSW SE is the development work that is provided by NETRONIC.

4.2 Architecture

The general architecture of the VSW SE is based on the following principles:

1. Other than in NETRONIC’s standard developer controls (such as VARCHART XGantt), NETRONIC will not manage data within the Gantt chart. All data management functions as well as the corresponding business logic happen within the customer application.
2. Nearly every mouse click on a graphical object (such as a Gantt chart bar or a resource row) in the VSW SE triggers a bi-directional communication between the VSW SE and the application (see “callbacks” and “Options/Methods” in the graph below).
3. This bi-directional communication leads to an exchange of objects as you can see in the graph below.


Website
Server part of application
Client part of application
Data objects
following data model
Options
Methods
Callbacks
Visual Scheduling Widget, Standard Edition
NETRONIC Web Application Framework
Developed by customer
Delivered by NETRONIC

All required objects are defined and identified within this FDD. These objects, the functions and the callbacks are part of a custom application programming interface (API) that can be addressed via interface.

4.3 Technological requirements

The VSW SE is developed based on HTML5/JavaScript program code. It can be performed using the respectively latest version of the following browsers:

- Google Chrome (Windows)

- Microsoft Internet Explorer 11
- Mozilla Firefox (Windows)
- Apple Safari (iOS, macOS)
- Microsoft Edge (Windows, upcoming Chromium based version only)

In case of automatic software updates of these browsers, NETRONIC will update its base technology accordingly. However, NETRONIC cannot guarantee an any-time availability for any browser version. It is recommended by NETRONIC to use Google Chrome; at the day that this FDD is created, this browser achieves – by far – the best performance results.

5 Visualization

In general the VSW SE can show two alternative Gantt Views: a Resource centric and an Activity centric View. Both views and its elements are described below.

5.1 Resource View

5.1.1 Overview

The Resource View can consist out of four elements for visualizing the respective data:

1. Table (Resources)
2. Timescale
3. Diagram area
4. Table (Entities)

This general structure is outlined below:

	2. Timescale	
1. Table (Resources)	3. Diagram area	4. Table (Entities)

5.1.2 Table (Resources)

The table area lists the resources. If a resource is sub-ordinated to another resource, it will be shown in an indented way in the table. The super-ordinated resource then also shows a symbol to collapse/ expand all its sub-ordinated resources. In case that a super-ordinated resource will get collapsed, the sub-ordinated resources will no longer get displayed. This enables a flexible nesting and grouping of different types of resources (e.g. employees to locations, replacement cars to garages, etc.).

The scope of VSW SE includes a one-column table only (e.g. to show one line of text which will be displayed to characterize the respective resource, incl. the above mentioned indentation and symbol to collapse/expand).

5.1.3 Timescale

5.1.3.1 Time range

The time range defines the date range that is shown in the current display of the VSW SE. This time range will be provided to the VSE as parameters from the application.

5.1.3.2 Timescale resolution

The horizontal resolution of the timescale can be adjusted by mouse wheel so that users can zoom in and zoom out of the displayed data in a seamless manner.

5.1.4 Diagram area

5.1.4.1 Visualization of resource calendars

The diagram area's background visualizes for each resource the working times (white) and non-working times (grey). This background calendar visualization cannot get changed interactively; it is just meant to display the general resources availability/ unavailability.

5.1.4.2 Visualization of resource allocations

The diagram area also shows the allocation of orders to resources as bars. The start and end date of a bar represents the time span in which the resource is allocated to that specific activity. If a resource has multiple allocations at the same time, the respective bars are shown one below the other.

Each bar can get a text to be displayed on it. The length of the bar determines the length of the text displayed (the text gets truncated at the end of the bar).

Allocation objects contain so-called entries that can be used to define phases (like startup time, runtime, teardown time). Each allocation entry gets a start time and an end time and can have a freely definable color. The entries are shown as sections within the allocation bar. An application can define the display of the bar (text and colors) flexibly and by their own via the respective data objects defined in the IDD.

Dependent allocations (predecessor/ successor) and their links will be shown as a curved arrow line between the predecessor and the successor. An application can control which links will be shown when.

5.1.4.3 Curves/histograms

For each resource, a histogram can be shown that visualizes both the capacity and the load of that resource. The histogram is displayed in the same row in the diagram in which the resource itself is shown. It can get expanded/collapsed by the user via a click on a symbol in the table area. The histogram will be applied to the same time area that is valid for the entire diagram area and its scroll and zoom behavior corresponds with the time scale. The histogram can include two type of curves.

1) Capacity curve
   The capacity curve is shown as line and visualizes the total capacity of a resource at a given time.
2) Load curve
   The load curve visualizes the utilization of this total capacity at a given time and is shown as filled area between the zero-line and the respective utilization.

The values of these curves are set independently from the resources' allocations. This means that they are not automatically calculated based on the resources' allocations. This enables the customer application to show these curves independent from the shown allocation and hence e.g. the “correct” allocation in case that some allocations had been filtered out.

In case the load curve exceeds the capacity curve, an overload exists. The overload area can get visualized by an own, independent color.

5.1.5 Table (Entities)

It is possible to show an additional table on the right side by option. This table can be filled with entity objects. If an entity is sub-ordinated to another entity, it will be shown in an indented way in the table. The super-ordinated entity then also shows a symbol to collapse/expand all its sub-ordinated entities. In case that a super-ordinated entity will get collapsed, the sub-ordinated entity will no longer get displayed. It is possible to drag and drop single entities from the table into the diagram area. This table can be used to show backlog activities.

5.1.6 Tooltips

It is possible to show a set of information that will get displayed when the user hovers with the mouse over a graphical element (tooltip text). A graphical element in case of the VSE is defined as

- Resource in the left table area
- Bar in the diagram area
- Links in the diagram area
- Entities in the right table area

An application can define the different information sets per graphical element by its own, and will be enabled to change these information sets later independent from NETRONIC.

5.2 Activity View(s)

5.2.1 Overview

The Activity View consists out of three elements for visualizing the respective data:

1. Table (Activities)
2. Timescale
3. Diagram area

	2. Timescale
1. Table (Activities)	3. Diagram area

Each building block of the VSW SE is defined as follows.

5.2.2 Table (Activities)

The table lists the activities. One activity can be the parent activity of other child activities. The super-ordinated activity is marked by a symbol in the table, which can be used to expand/collapse the underlying activities. If an activity is sub-ordinated to another activity, it will be shown in an indented way in the table.

The scope of VSW SE includes a one-column table only (e.g. to show one line of text which will be displayed to characterize the respective resource, incl. the above mentioned indentation and symbol to collapse/expand).

5.2.3 Timescale

The timescale is exactly as specified in the Resource View.

5.2.4 Diagram area

The diagram area shows activities as graphical objects. There are three different activity types resulting in three different visualizations:

Summary bar. The summary bar is typically used to visually embrace all subordinated activities. It spans the period from a given start date to a given finish data and will not be calculated automatically.

Activity. Activities are shown as bars. The time between the start and end date of a bar represent the duration of the activity. Each bar can have a text displayed on it. Also, each bar can have a freely defined color. The flexible creation of the bar display (text and color) can be done by an application via the respective data objects. The length of the bar determines the length of the text displayed: the text gets cropped at the end of the bar. It is also possible to define a percentage of complete for each activity, and to have an additional visualization of it.

Diamond. A diamond (often called milestone) is shown as a point of time symbol.

Dependencies between activities are shown as links between the bars.

5.2.5 Tooltips

It is possible to show a set of information that will get displayed when the user hovers with the mouse over a graphical element (tooltip text). A graphical element in case of the VSW SE is defined as

- Activity in the table area
- Bar in the diagram area

6 Functionality: Drag & Drop

The Resource View of the VSW SE visualizes allocations of resources and allows the user to change these allocations interactively via drag & drop. The same applies to the activities in the Activity View. In the following chapters the supported graphical user interactions are described.

6.1 Selection

6.1.1 Click

The user can select a resource, an allocation, an activity, or an entity.
This also triggers the callback o*nSelectionChanged* which passes the ID of the selected object. This allows an application to react to this selection within the customer application accordingly by e.g. displaying additional information in another window.

6.1.2 Double click

The user can also double click on a resource, an allocation, an activity, a link or an entity. This triggers the events *onDoubleClicked* accordingly including of the respective ID of the double clicked object. An application then can react to these events e.g. by opening a dialog for editing the selected objects.

6.1.3 Multi selection

The user can also select multiple resource allocations of resource with a left mouse click. This can be done either by CTRL + left mouse click or by pulling up a selection rectangle. Then all allocations get highlighted and the respective callback is triggered.

6.1.4 Right-click / context menu

The user can get enabled to open a context menu upon the right click on resource, an allocation, an activity, a link or an entity. The respective fitting context menus can be freely defined and associated with these elements by an application. If the user opens a context menu and selects an item, the callback *onShowContextMenu* gets triggered. This event passes the object type and the ID of the respective object to the application.

6.2 Movements: Change of allocations and activities

The following changes of allocations are allowed:

- Vertical movements = change of allocation from one resource to another (see above Visualization of resource allocations)
- Horizontal movements = change of start and date dates of an allocation
- Change of length of bar = change of duration of an allocation

The following changes of activities are allowed:

- Horizontal movements = change of start and date dates of an activity
- Change of length of bar = change of duration of an activity

All movements can get done with time-spot accuracy (on minutes- or even on seconds-level).

Once the user completed a movement, the VSW SE triggers the callback *onDrop* and passes the respectively changed data to the application. This callback is a request to change the data accordingly. An application can then react to this request with its own business logic, which then decides if the user's request gets approved, rejected or modified. In that way, the application can trigger an update call to the VSW SE which will then carry out the confirmed changes.

6.3 Dragging entities

If the entities table is activated the user can schedule unscheduled activities by interaction. This is done by dragging a selected row of the entity table into the diagram area. This can be used for instance to assign unassigned backlog activities to resources.

6.4 Creation of objects

NETRONIC will not provide any functionality to allow the interactive and visual creation of a new object including the respective resource allocation. NETRONIC recommends that an application implements this functionality via buttons, which will be shown in the same windows that contains the VSW SE instance.

6.5 Deletion of objects

NETRONIC will not provide any functionality to allow the deletion of objects by user interaction within the VSW SE. An application needs to implement this functionality by providing context menu items or buttons.

7 API

Changes to the VSW SE can not only be executed via the user interactions specified in 6. In addition to this, they can also get triggered via program code calling API function.

There are additional functions to the add/update/remove functions related to the data objects, which can get applied to load and change data in the VSW SE. These additional functions can get used to

- Reduce the diagram area to a certain time interval
- Scroll horizontally to a specific time within the time area
- Scroll horizontally and vertically to a specific allocation/activity in the diagram area
- Blend out all non-working periods, which all displayed resources have in common

8 Functionality not included

As the VSW SE is a jQuery UI widget and as printing from a web-based environment often causes unwanted effects, NETRONIC will not provide any functionality to print the contents of a VSW SE instance.

9 Document change history

Version	Comment
1.0	Initial Document
3.0	Changes for VSW SE 3.0