



New Features Overview

(a.k.a Smaart v9 for v8 users)

This Guide – Start Here

This guide is a quick reference to acquaint you with what is new in Smaart v9 and is written specifically for users upgrading from and/or familiar with Smaart v8 or previous versions.

If you are new to Smaart or have limited experience operating dual-channel FFT-based measurements, we would strongly recommend also consulting the current Smaart v8 User Guide and other support materials available at support.rationalacoustics.com, as well as the program's extensive on-board help files (found in the software's "Help" menu).

While Smaart software can certainly be described to exist in a state of continuous development, the work that comprises the v9 upgrade cycle represents a significant amount of effort in and of itself - almost 3 years of focused development time (in the midst of a pandemic no less). With the initial "9.0" release, you will find that a good deal of the measurement functionality and operational paradigms to be much the same as those of previous versions – although with some interesting changes/additions. This is specifically done so as not to throw away progress already made from past development, or to invalidate any skills or capabilities we as operators have learned over time.

Much of the development work in this upgrade cycle was dedicated to structural, "under the hood" improvements. Coding work that was far too involved, too foundational and/or time consuming to undertake within our standard ongoing "update" cycles – work that may not be initially obvious, but the stuff that makes the future of the platform possible. This document takes you on a tour through all of that which v9 development comprises, the obvious and the less so. Of course, we know our users (and ourselves) and many of you are already diving into the software, or already skimming looking for graphics (this is already the 4th paragraph, after all, so congratulations for reading this far). If that is the case, we highly recommend at least jumping to the last two sections of this document, "[Refine](#)" and "[Enhance](#)", which will help you get your bearings for navigating the new features and paradigms in v9. You can catch up on the rest of the document later.

Also, please be aware, while the initial release version 9.0.0 includes the features and functionality detailed here, new v9 "update" releases will be published regularly over the coming weeks and months to add new features and mature the platform, as well as to address the inevitable "undocumented features" (a.k.a. bugs). These "after dot" updates are free to all registered v9 licensees and can be accessed from within the software or from your license account at my.rationalacoustics.com. These periodic releases will be announced on our website and via our newsletter (which if you have not already done so, we would highly recommend signing up for on our site.) Moreover, all licensed v9 Smaart users are invited to participate in beta testing new features as they are developed, prior to their official release. These "Beta builds" are also available from your license account.

One final note, please feel free to contact Rational Acoustics at support@rationalacoustics.com for help with any Smaart issues, to report software bugs or to suggest/request new features and functionality for future updates.

Table of Contents

This Guide – Start Here	2
Table of Contents	3
Smaart Version 9	5
Updates vs Upgrades – the Smaart Development Process	5
Updates	5
Upgrades	5
The v9 Upgrade – the Why and What	6
Refresh / Rebuild – Strengthening Our Foundation	6
The Refresh/Rebuild Bullet List	8
Restructure – Adapting Smaart to Our Users/Markets.....	8
Subscription Licenses	9
Smaart’s New Edition-Based Product Structure.....	10
Complexity vs Usability?	10
Smaart Editions	11
Refine – Making What We Have Work Better	12
Separation of Measurement and Graphing Controls.....	13
Changes to the Menus.....	13
Preferences and the “Advanced” Tab.....	14
GUI (Graphical User Interface) Changes	14
View Controls	15
Controls in the Plots.....	15
Graph Settings	15
Smoothing / Banding	15
Weighting.....	15
Add/Subtract Plot	15
Axis Labels	16
Mode Buttons	16
Reworked Configuration Flow	16
Measurement Engine Singularity.....	16
New Measurement Config Dialog	17
Drag and Drop.....	17
Phantom Engine	17
Engine Right-Click Menu	18
New Engine Detail.....	18
Updated Measurement Controls	18
Tracker Function	18
Pause	19
New Group Measurement Controls	19
New TF Averaging Depth Setting	19
IR and SPL Mode Updates	20
Enhance – Adding New Capabilities.....	20
Enhanced Data Structure.....	20

WHAT'S NEW IN SMAART 9.0

Larger Live FFT's Available in RT mode	21
Single FFT and MTW Always Calculated	21
Use MTW Coh Option	21
MTW+ Replaces old MTW	22
Expanded Output Capabilities	22
Output Presets	22
Timer (Macro) Function.....	23
Parting Reminders	23

Smaart Version 9

Updates vs Upgrades – the Smaart Development Process

The Smaart software platform has been in constant development since 1995 – over 27 years! (Longer than some of our users have been in development.) Over time, this dev process has taken the form of two distinct modes: (minor version) Updates and (major version) Upgrades.

Updates

The minor version **UPDATES** – the “numbers after the dot” releases – are published periodically throughout the year and constitute the ongoing feature additions, refinements and reworks that come as we expand and refine our applications and processes for using this tool. Commonly referred to as the “dot releases,” this continual update process is critical for maintaining the code, addressing bugs and adapting to changes in the hardware and Operating Systems we depend upon.

The update process uses relatively short development cycles which comprise their own major and minor release components. The main update releases, the first number after the dot (i.e., v9.X) generally take upwards of 6 to 9 months of development and testing to produce (although some have taken significantly longer), and generally contain a few major feature additions and a whole bunch of minor ones. Each of these major updates is normally accompanied with an update to the software documentation – the program help files, the User manual and support materials – and is always announced on as many communication channels as we can (see previous note about signing up for our newsletter.)

The minor update releases, the “second dot” or the “number after the dot after the number after the dot” (i.e., v9.1.X), come at a more frequent and slightly erratic pace, as they usually comprise bug fixes and minor feature tweaks. These updates release as necessary with very little fanfare - so before reporting a bug, check to see if an update has already been published (*pro tip*).

During the development process for updates, licensed Smaart users can actively take part in the beta testing of new features. Users who wish to participate can download the current beta versions from their Smaart license accounts and can actually have both the released and beta versions installed and usable under their same license.

Updates are offered at no cost to licensees of the current version and can be automatically checked for and loaded from the software if you have an active internet connection while running the program.

** Of special note to 9.0 users, the initial three months of the major release is called the “early adopter” period and represents a time of significantly rapid update releases.*

Upgrades

The major version **UPGRADES** – the primary “number before the dot” releases, the “version number” – happen on a less frequent basis (5- or 6-year cycles) and constitute a much more significant undertaking. While the update process modifies and expands the existing codebase, the Upgrade build generally addresses structural issues on a deeper level and represent a major overhaul and rebuild of the

codebase and its underlying development environment. Whether this is to address changes in underlying technology, operating systems, or addressing previous limits due to our own past coding decisions (self-inflicted limits?) the version upgrade process normally involves breaking everything and then putting it back together - hopefully fundamentally changed and prepared to take on the future.

With Smaart, Upgrades of standard, “perpetual” licenses are a paid cost. This is how the continued support, maintenance, and development of the platform are funded. In the case of subscription licenses, these costs are translated into a yearly ongoing fee, and no specific Upgrade cost is required.

Owners of previous version perpetual licenses are not required to upgrade to the new version – your old version will continue to operate as before. However, once a new version has been released, all new feature development for Smaart is focused only on, and offered only for, the current version platform. We actively develop and maintain only one version of Smaart at a time. **No new updates will be released for the previous version.** Rational Acoustics does take seriously our commitment to supporting your old license’s use and installation – in other words, we will still help you with maintaining installations of your old version licenses for as long as they will operate under your computer’s current OS. But if a new OS breaks your old version, we will not be spending development resources to fix it (maybe tread lightly when updating your OS’s if you plan to keep running old software?) Also, Rational Acoustics will not be offering sales of new license seats for old versions. For further information regarding these policies, consult our website at RationalAcoustics.com, or email us at sales@rationalacoustics.com.

This release, Smaart v9.0 is an Upgrade release. Let’s dig into the details of Why we needed to build it, and What we did.

The v9 Upgrade – the Why and What

Updates are driven by the desire to add functionality and refine operation. Upgrades are driven by the fundamental need to change the structure of the program to make Updates possible, now and in the future. There are four fundamental drivers for the v9 upgrade:

1. Refresh/Rebuild – our codebase and our development tools and environment
2. Restructure – our products to better fit our markets and our users’ requirements
3. Refine – the continual need to refine and evolve the Smaart measurement environment
4. Enhance – our measurement capabilities and data structures to allow for cool new stuff

Refresh / Rebuild – Strengthening Our Foundation

Over 27 years of development, this part of the Upgrade process has truly proven critical. We do not create the environments and tools we use to create and execute the code that is Smaart software. Because we have chosen to offer Smaart as software (which provides many benefits to the user and the product), instead of as a program/firmware operating on locked-down hardware with a fixed OS, we have chosen to build a house on an ever-shifting foundation. When Smaart was first developed, it was designed to run under Windows 95 on 33 and 66 mHz 486 and Pentium processors! After 12 years of development and 5 major versions - Smaart, Smaart Pro, Smaart Pro 3, SmaartLive 4, SmaartLive 5 (we don’t talk about 6) - the program was an industry standard, but its codebase was a hideously tangled, inflexible, single-threaded mess. Pushing Smaart forward required a serious rethink. In 2008, after founding Rational Acoustics, we switched development environments and completely rebuilt Smaart

WHAT'S NEW IN SMAART 9.0

with a modern, object-oriented architecture that could make use of the modern multi-core CPU's and multi-channel IO devices on multiple OS's.

This v9 development cycle, as v8 had before it, began with the essential task of updating our development environment, its tools and of course the FFT libraries we rely on, to assure that Smaart would be **fully compatible with current 64bit operating systems and CPU chips (like the new Apple M1 and M2 processors)**. In effect, before renovating the house, we started by shoring up the foundation.

We updated our development environment. Our compilers. Our dev computers. How we stored and updated our source code. How we tracked our work. We took the time to rethink and retool how we did everything.

One nice benefit of this work is that the IO devices that Smaart relies upon have now become truly **“plug-n-play.”** Up until now, Smaart would only discover connected IO devices on boot-up. This is why standard practice has been to connect your IO device first, then start Smaart. Now if you forget to connect before starting the program, no problem – just plug it in and Smaart will find it, no need for re-boot.

Also, speaking of re-booting, one of the great side benefits of the code refresh and rebuild is that Smaart now **boots significantly faster and loads configurations (or restores defaults) in seconds**, and without requiring a program re-boot.

It also gave us a second chance to remake some fundamental decisions made early on that are limiting us now. Things like, limiting Smaart's generator output to 2 channels (there were only two possible when we made that decision). Or reinitializing the audio drivers whenever we made a major (or minor) program change – this is why Smaart's generator pauses sometimes when you are doing stuff that has nothing to do with the Gen like changing the plot layout (?!?). Why did we do this – I guess it made sense then, so we've had to live that nonsense ever since.

The v9 Rebuild & Refresh provided an opportunity to address a significant list of past coding decisions and “solutions” like these – issues that were so deep in the program that changing them would take more time than we wanted to give at the time and messing with them might break a bunch of other stuff too, so we decided it was better to live with them and not poke the bear. Well, since we were in there breaking everything else anyway, now was the time to go after them.

However, by far, the most arduous/daunting task of this Upgrade “Refresh & Rebuild” process was **updating our rendering systems**. A major part of the Smaart software involves plotting massive amounts of data in real time. Oftentimes, when users were maxing out their computer resources, it wasn't the CPU (Central Processing Unit) that was the limiting factor – it was the GPU (Graphics Processing Unit). We are often plotting many multiples of data traces, with upwards of 16,000 complex points apiece (ok, yeah, 16,384 for the geeks out there) into plot areas as small as 480 x 190 pixels! Somebody has to reduce that data and make it fit, and that fell on the GPU. Put simply, the more data we wanted to sling, the more attention we needed to pay to our rendering systems.

In 2008, when we began the first “Rational” rebuild of the Smaart code to create Smaart v7, we chose to use OpenGL for our graphics rendering. This was a great choice at the time because we were also beginning to develop for multiple platforms (Windows and Mac OS), and OpenGL worked great for both. Fast-

forward to the 2020's, and our rendering requirements have continually increased, but OpenGL is now being deprecated* in macOS, and gently put out to pasture in Windows.

*DEPRECATION, IN ITS PROGRAMMING SENSE, IS THE PROCESS OF TAKING OLDER CODE AND MARKING IT AS NO LONGER BEING USEFUL WITHIN THE CODEBASE, USUALLY BECAUSE IT HAS BEEN SUPERSEDED BY NEWER CODE. (THANKS INTERNETS)

So yeah, this meant our current OpenGL-based code would still run, just less and less well over time. This left us with the awesome “opportunity” to completely rewrite Smaart using THE new, modern, massively capable, preferred rendering engines for our target OS's – Direct X for Windows and Metal for Mac. To be honest, it is a significant improvement and does provide some great benefits (like iOS uses Metal – so creating apps would be easier), but so does rewiring your whole house's electrical system and bringing it up to code. Beneficial, yes, but definitely not a weekend project. And we are certainly still on a learning curve with our rendering right now.

In summation, we broke and rebuilt Smaart – and to be honest, there may be a few things left to be rebuilt in a dot release, but the result is a clean, stable, refreshed codebase that will allow us to build and maintain Smaart into the future (when we all will probably be wearing one-piece body suits and living in bio-domes or on starships).

The Refresh/Rebuild Bullet List

So, here is the dev environment Refresh/Rebuild part of the v9 Upgrade process in an “easy to skim” bullet list: (AKA – “What did our programmers do for their first year of the pandemic”)

- Updated our Dev environment and tools
- Made sure we were fully 64bit OS compatible
 - And Apple silicon (M1 / M2) compatible
- Fixed some deeply imbedded past bad coding decisions
- Faster program boot and configuration loading
- Actual Plug-n-play for IO's
- Updated our rendering systems to DirectX and Metal
- Re-Built our licensing systems to add Subscription Licenses (See “Restructure” below)

Restructure – Adapting Smaart to Our Users/Markets

Another of the primary goals of the v9 Upgrade cycle was to rethink, and as needed restructure, our Smaart product offerings to address our users and markets. Over ten years ago, we began offering the ability for users to expand their licenses with additional “seats.” It was the natural response to business' requests to add more installations without having to manage multiple separate licenses and accounts. Soon after we added the ability for users to purchase and manage license seats themselves directly from within their Smaart license accounts.

As it turns out, license seats were extremely helpful and a big success, particularly with larger companies who were managing and fielding a significant number of Smaart rigs. With the advent of the v9 Upgrade

development cycle, we again asked ourselves “Are we making Smart products that fit our current users’ needs?”

Our answers led us to undertaking two initiatives:

1. Offering subscription licenses
2. Creating a revised, “Edition-based” Smart product structure

Subscription Licenses

Up until v9, Smart software has always been offered only as a perpetual license, where users pay once for their initial version license - either as a new license or as an upgrade from a previous version – and then any updates to that version are made available at no additional cost (see previous section “Updates vs Upgrades” for more discussion). The software itself basically continues to function in perpetuity – at least until it is no longer supported by current operating systems. This is the classic model where the software purchase is directly analogous to buying a piece of gear, and it is the model that software companies have used for years to pay for the ongoing development, maintenance and support costs associated with offering a software product.

Over the past decade however, made increasingly possible by the ubiquity of internet access, another software licensing model has gained favor with software companies and users – the subscription license model. Here the user pays for access to the software via a periodic (yearly or monthly) subscription charge. When the user stops paying their periodic fee and the subscription lapses, the user loses access to the software. This licensing model does require that the computer running the software have at least periodic access to the internet for subscription verification (sometimes a deal-breaker for mission-critical show computers), however in most use cases, this does not present a significant barrier to viability.

On the positive side, the subscription license model provides a few unique benefits to both the user and the software company:

- The user always operates on the current software versions (no upgrade/update costs)
- The initial cost of access to the program is significantly lower
- Subscription fees are consistent and budget-able for the user and the software company

On the negative side:

- When the subscription runs out – user loses access to the software
- Program must have periodic internet access
- Initial and Ongoing (Version Upgrade) costs higher in the short term

In many ways, it is like the difference between purchasing and leasing a vehicle.

Over the past decade, as subscription models for software have become more common, Rational Acoustics has consistently heard two comments from our Smart users.

1. “Please do NOT go to a subscription model!” – very common particularly early on
2. “Can we license Smart as a Subscription” – becoming more and more common

The answer to these statements/questions is **YES** – Smart v9 will be available via subscription license **AND** as a classic perpetual license. One or the other - choose the licensing model that best fits you.

One of the biggest projects during v9 Upgrade project has been the rebuild and expansion of Smart's licensing system to allow for subscriptions. For more information on Smart subscriptions as well as general license pricing, please consult the [Rational Acoustics web site](#).

Smart's New Edition-Based Product Structure

Complexity vs Usability?

As we teach in our fundamentals class, Smart is simply a tool. Albeit a complex one that sometimes requires advanced understanding to wield, it is, in its essence, a measurement tool like a voltmeter that an audio engineer uses to ask and answer questions as they make decisions and perform their jobs. The engineer is the analyzer – they define/configure their measurements (ask the questions), read and interpret the data, and ultimately decide their actions based on its data and their knowledge of their task. Smart's job is to perform the measurements and feed the resulting data to that analyzer (engineer). A great deal of Smart's feature set is specifically focused on prepping and presenting its measurement data in a form that is usable by the engineer – to provide actionable data.

As we have previously stated, Smart has been in active development now for almost three decades. Over that time, it has consistently gained in features and functionality. Often that increase in capability has also come with a corresponding increase in complexity - which in turn has sometimes decreased its usability as a tool for some engineers. And here then is the primary conundrum in building a measurement tool like Smart, does an increase in capability result in a decrease of usability?

A great deal of development effort has been spent over the course of Smart's growth to balance that trade off. A lot of work has gone into adapting the software's GUI to hide complexity at times to keep it from interfering with common workflow, but to make it readily available when it is needed.

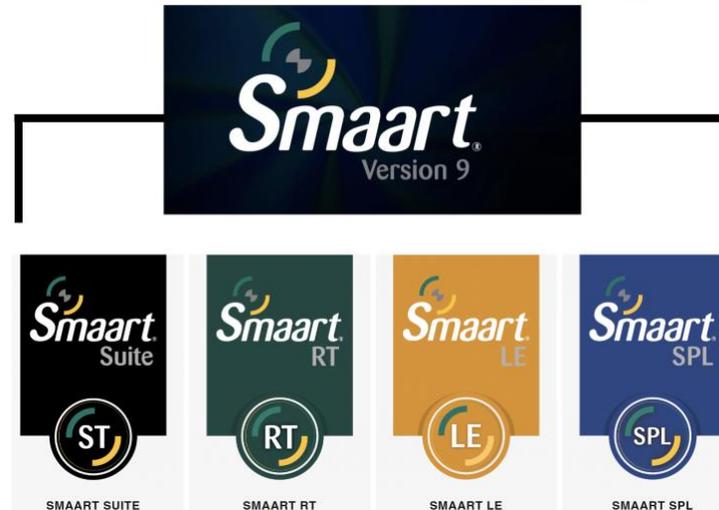
The balance between capability and usability is an important concept here – to be a powerful tool, it has to be usable. Many Smart users operate it in environments under *very* tight time constraints. “Unwieldy (slow) but powerful” may often equate to simply “unusable.” There are many, very experienced users, “power users,” that only use a fraction of Smart's capabilities and functions. There are others who couldn't do their jobs without some advanced features that most users might call esoteric. Usability and power then are all about the user, and what they need, and less, maybe, about raw feature count.

Over the past 5 years, Rational Acoustics has dedicated a large amount of development time to significantly enhancing both Smart's IR mode and SPL mode. That work represents a great deal of added capability – and yet, there is a large percentage of our userbase that never uses either of those modes, and others that use only one of them exclusively (like users who only use the SPL mode, but none of the measurement and analysis capabilities in Real Time and IR modes.)

The primary question for a user then becomes, “What is the right feature set and complexity for me?” Or to look at it from the perspective of someone purchasing a license, “Why should I pay for features and functionality I never intend to use?”

WHAT'S NEW IN SMAART 9.0

To help answer these questions, Rational Acoustics has adopted an Edition-based product structure for Smart software.



Smart Editions

With the release of v9, Smart is moving from its old version-based product structure to a new, more cohesive edition-based one. This means, instead of maintaining multiple separate products, with separate but related code bases, each at different release versions (i.e., Smart v8, Smart Di v2, Smart SPL v1), there will now be only one master version of Smart (v9) that is available in several different editions.

By standardizing all editions of Smart on the same version code base, we can ensure that all editions share the same fundamental GUI, the same command and control structure, operational paradigms, and are cohesive down to the hotkeys. The only difference between the editions is their feature sets. This means, learning to navigate and control one edition should translate to all.

The 4 editions of Smart available with the release of v9 are:



Smart Suite is the full-featured, flagship, edition of Smart v9 which includes all three of Smart's measurement modes (Real-Time, Impulse Response, and SPL) with no functional limitations. Smart Suite is the analog to the comprehensive functionality currently found in Smart v8 and is designed for professional users who require (or desire) access to the full feature set available across all three of Smart's measurement modes. Basically, it's everything in v9.



Smaart RT (Real-Time Edition) is Smaart Suite's Real-Time Mode as a stand-alone edition. It comprises all of the core multi-engine FFT-based measurement capabilities that professional audio engineers have come to rely upon from Smaart over the past 27 years – RTA, Spectrograph, Transfer Function and Impulse Response measurements, all harnessed in a work environment developed to fit the workflow and task critical demands of modern audio engineering. If you hear an engineer talking about using “Smaart,” they are probably talking about the RT mode. While it does include IR measurement and basic single-channel SPL monitoring capabilities, Smaart RT does not include Suite’s acoustic analysis focused Impulse Response Mode or the fully featured SPL monitoring tools of SPL Mode. Smaart RT is designed for users who simply require the complete advanced functionality of the Real-Time Mode.



Smaart LE (Light Edition) is a simplified, streamlined version of Smaart's Real Time Mode with many of the advanced settings, adjustments and tweaks for the Spectrum, Transfer Function and IR measurements pre-set and fixed – in effect, just like operating RT on its default settings. All the power of RT without all of the variables. While it does not include many of the advanced user functions included in Suite and RT, it does still allow for Smaart’s unlimited multi-engine measurement configurations. Moreover, because it uses the same GUI and command and control structure as RT and Suite, it serves as a great gateway to the Smaart measurement platform. Smaart LE contains the core, essential functionality for Smaart’s spectrum and frequency response measurements and is a great choice for users just starting out on the Smaart platform, students, and those who want quick, simple, solid measurement functionality without the bells and whistles.



Smaart SPL (SPL Edition) is the power of Smaart Suite’s SPL mode as a stand-alone edition. Smaart SPL includes all of the multi-channel, remotely monitorable, SPL metering and logging features found in Smaart Suite, in a simplified, purpose-focused package. Often a perfect solution for installed level monitoring systems, Smaart SPL is ideal for applications that don't require any of the Real Time or Impulse Response measurement functionality found in other editions of Smaart.

Refine – Making What We Have Work Better

A significant portion of the development work in all Smaart update and upgrade cycles is focused on the continued evolution of the established measurement environment. The extension and refinement of the current version, its functionality and operational paradigms. How can we now take it further – improve what we have, make it easier to use, more intuitive – less clunky?

Moreover, how can we flatten the learning curve for new users, make Smaart more intuitive and easier to learn – and relearn. After all, a large portion of Smaart users are not using it every day, or even weekly or monthly. For many, it’s a tool in their toolbox that they take out when they need it – but there can be long stretches of time between uses.

When considering the refinements, we want to make in our upcoming development cycle, we ask ourselves a series of questions:

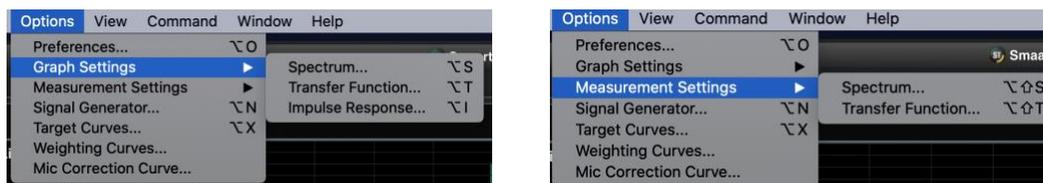
- Are controls and information where they need to be? Are they where you would expect them to be? Is their placement intuitive.
- What functions are useful but are non-intuitive to operate.
- How are features actually used as opposed to how they were originally designed to be used?
- Does the work process using a feature flow smoothly, or is it clunky?
- What are functions that you use frequently that take too many clicks to access or activate? Can we get to that function faster?
- Are the tools in place for you to adapt Smart to your workflow?

Many of these questions may seem obvious, but it is surprising how often we adapt to how features function instead of adapting the feature function to how we need it to work.

The following section documents the changes and “refinements” that have been made to **Smart Suite and its three modes** in V9. All functionality from the previous version Smart v8 has been retained - except where noted. Controls and control placement for some functions however may have changed.

Separation of Measurement and Graphing Controls

While virtually all functionality and controls have been retained, a specific effort was made to separate the controls that effect the actual calculation of the measurement data from those that effect the display/graphing/presentation of that data. For example, controls such as FFT size or Averaging Depth actually change the measurement, whereas trace Smoothing or spectrum data Banding simply change how we view the data. While Smart has historically conflated these two types of controls, in v9 we attempt to separate them and their controls appropriately. This organizational change can be seen in v9 with the separation of the old Spectrum and Transfer Function options dialogs into separate Graph Settings and Measurement Settings dialogs for each.



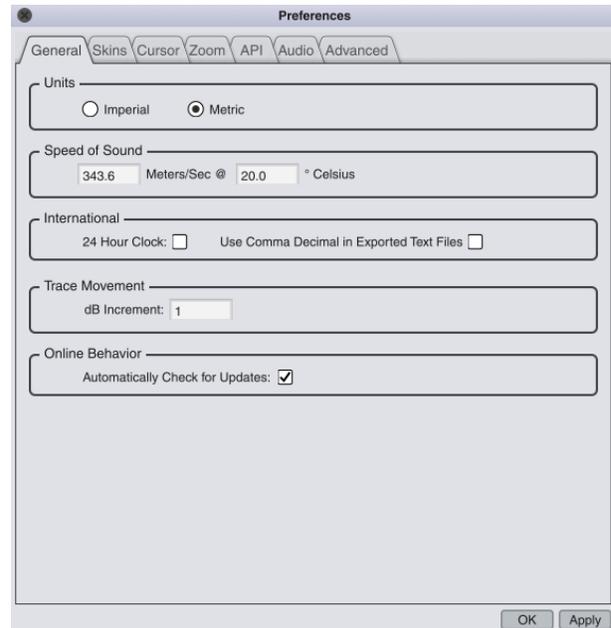
Access to the Measurement Settings dialogs can be gained from the measurement type label at the top of the Control bar, via hot key, via the options menu, or via a button in the lower left Graph Settings dialog. Likewise, the Graph Settings dialog can be accessed via the “Cog” button located in the lower right of the plots, via hot key, via the options menu, or via a button in the Measurement settings dialog.

Changes to the Menu

On the whole, the menu structure in V9 remains mostly the same as it was in v8. The biggest changes can be found in the Options menu with the Graph and Measurement settings dialogs discussed above, and the addition of a Tabbed Preferences dialog

Preferences and the “Advanced” Tab

The Tabbed Preferences Dialog houses many of the Options dialogs that previously populated the main v8 options menu. The dialogs that have been placed in the new Preferences tab are basically ones that effect general program settings that are usually set once and forgot, or more often, simply left at default. This dialog also houses the “Advanced” tab. Originally named the “TBD” tab (There Be Dragons) during development, the purpose of this tab is to hide advanced functionality that could be confusing or distracting to users who do not understand the features. As history has taught us, many (most?) of our users will activate a feature/control even if, or specifically because, they do not know what it does. This in turn becomes a support liability relatively quickly. To decrease the temptation, controls enabling some advanced functionalities are and will be housed here. (Much of the planned advanced functionality that is planned for v9 has not yet been published in a released or beta version.)



GUI (Graphical User Interface) Changes

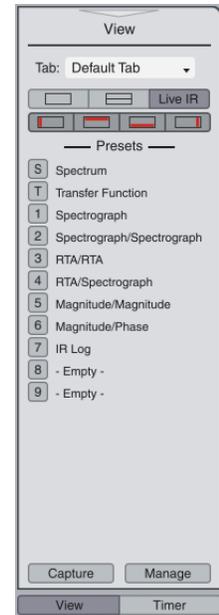
At first glance, the GUI for v9 is very similar to that of v8, particularly in general layout. In v9, the main GUI is divided between five main areas: (clockwise from the left) the Data Bar (left), the Tab/Title Bar (top), the Control Bar (right), the Command Bar (bottom), and the Plot Area (center). As in v8, each of the four “Bars” surrounding the plot area can be hidden to allow more room for the plots. The Data bar can also be widened to expose more area to display longer data names.

One change from previous versions, the indication for which data plot has operational focus is now achieved with a simple “highlight” border around the plot with focus. As before, the user can adjust the colors used for the GUI in the “Skins” options located in the Preferences tabs.

On closer inspection, a few significant changes to the GUI in v9 become obvious:

View Controls

Critical to an engineer's ability to read and analyze measurement data is the ability to set and vary the layout of the plot areas and program bars on the screen. In the past, the primary way of controlling the plot layout in Smaart was via the preset view controls in the View menu and by memorizing and using the hot keys for triggering them. Smaart v7 and v8 also had buttons in the Control bar that triggered some view functions, but they were far from comprehensive controls, and in many ways, a bit confusing. With most users, those controls were primarily used for triggering either the standard "Spectrum" view (a single RTA plot) or the standard "Transfer Function" three plot view (Live IR over Phase over Magnitude). In v9, those view control buttons have been replaced with a much more comprehensive View control, pop-up area. This control area brings together much of the view control functionality that was spread throughout the menus and GUI into a single control area. Of course, it contains the old controls as part of it, and all the old menu-based and hot key-based view controls are still in place. In fact, two of the buttons on the command bar have been assigned to triggering the Spectrum and TF views, just in case you missed them from the top level of the GUI.



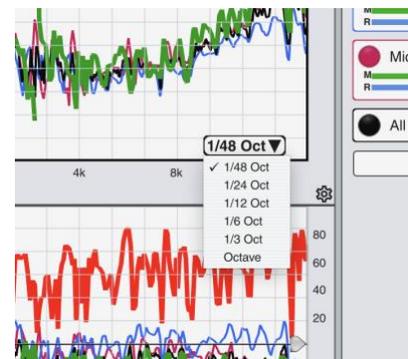
The View control Pop-up is accessed from the View button on the Control bar just above the generator controls. Clicking on the view button again, or outside of the pop-up, will lower and hide it again.

Controls in the Plots

In order to make controlling Smaart more intuitive, particularly for newer users, a significant effort was made to put controls near what they controlled. This became particularly useful with the functions that controlled how we viewed our data.

Graph Settings

Access to the Graph Settings for a plot type (Spectrum or TF) can be accessed via the "Cog" icon in the lower right of the plot area.



Smoothing / Banding

Smoothing and Banding controls are also located in the lower right of the plot within the actual graph area.

Weighting

Weighting controls, once weighting has been applied to a trace, appear in the upper right of the plot, directly below the Trace name and dB offset value.

Add/Subtract Plot

Adding an additional plot to a single-plot layout or subtracting a plot from a two-plot layout was previously a bit of a convoluted process with the old view controls. Now adding and subtracting plots is straight-forward using the in-plot controls in the lower left corner of the plot area.



Axis Labels

Ok, this one was way overdue. Somewhere during the process of building v7 we stopped labeling the Units on our plot axis (i.e., Frequency (Hz), or Level (dB)). This continued through v8, but they are now back in v9.

Mode Buttons

Because it has all three measurement modes (Real-Time, Impulse and SPL), Smart Suite's GUI has mode buttons in the upper right corner of the main GUI to select the active mode. An option in the Preferences: Advanced dialog sets these controls to create a new window for each mode when it is accessed via these buttons so that a user can operate multiple modes simultaneously.



Reworked Configuration Flow

Measurement engine configuration and management has once again been refined in v9. Refining this functionality, because it is so fundamental to defining and managing our measurements and workflows, makes it one of the key areas that we focus on in the upgrade process.

Measurement Engine Singularity

The most significant change in the configuration process is the adoption of what we call "engine singularity." This is a formal way of saying that once you create/define a Spectrum or TF measurement engine, it is available for use in any work tab within the program – you don't have to create clones of that engine for use in separate Tabs.

While this may sound like an obvious way of working with multiple measurement engines, the reality of how engine management has worked in Smaart in previous versions has not been so straightforward. When we created Smaart v7, we had never worked with managing the configuration and operation of multiple measurement engines run simultaneously. Our first attempt at a solution involved creating **Groups** of TF and Spectrum measurement engines that the program could switch between. (This is why the hot key for accessing the measurement configuration is still Option + [G]). While functional, this system proved to be extremely clunky and time-consuming to configure.

When v8 was introduced, a major enhancement was the addition of Tabs to manage workspaces and measurement engines within Smaart. This change represented a huge leap forward in usability and configuration speed. However, because the "Measurement Groups" concept had been so fundamental to the v7 code, the choice was made during the v8 Upgrade development to modify the measurement groups to create v8's tabs. The resulting byproduct was, that while easier to configure and manage, using the "same" measurement engine in multiple tabs still required creating and managing clone engines – effectively, the measurement engines were still "owned" by the Tab (Group) structure. Deleting a Tab effectively deleted the engines within it.

In undertaking the current Upgrade, we have finally dug down and undone our measurement engine's existential attachment to the Tabs (Groups). Now in v9, you configure engines and choose to use them or not in your active Tab. Once you create them, they are available in the program until you choose to delete them. Deleting a tab has no effect on the configured engines. Sounds simple? In concept, yes, that's the point. It just that getting here was a rather circuitous path.

New Measurement Config Dialog

V9 features a new Measurement configuration dialog that expands out from the left of the Control Bar. The new Measurement Configuration dialog does not have Measurement settings

Measurement Configuration	Device	Meas. Ch.	Ref. Ch.
<ul style="list-style-type: none"> [-] Spectrum <ul style="list-style-type: none"> mic 1 mic 2 mic 3 gen direct galaxy 6 out [-] Transfer Function <ul style="list-style-type: none"> Mic 1 TF Mic 2 TF Mic 3 TF Galaxy 6 [-] Transfer Function Live Average <ul style="list-style-type: none"> All Mic TF 			
	Dante Virtual Soundc...	mic 1	
	Dante Virtual Soundc...	mic 2	
	Dante Virtual Soundc...	mic 3	
	Dante Virtual Soundc...	gen direct	
	Dante Virtual Soundc...	galaxy 6 out	
	Dante Virtual Soundc...	mic 1	gen direct
	Dante Virtual Soundc...	mic 2	gen direct
	Dante Virtual Soundc...	mic 3	gen direct
	Dante Virtual Soundc...	galaxy 6 out	gen direct

+ + +
New Spectrum Measurement
New TF Measurement

+ + +
New Spectrum Average
New TF Average

(those are in the “Measurement Settings” dialogs) or Tab information. It simply details the currently configured engines and allows you to add or delete them or access the individual engine detail tabs by double-clicking on a given engine. Access to the Measurement Configuration dialog gained via the Config Menu, hot key (Option+[G]), or with the Measurement Config button: 

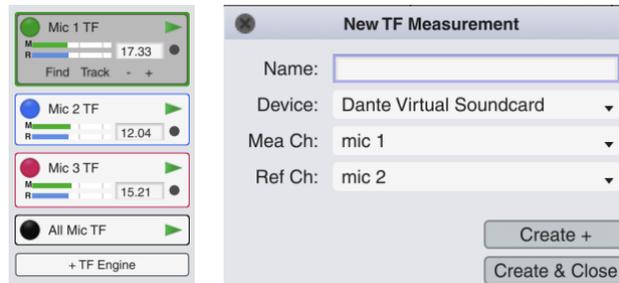
Drag and Drop

Two cool new applications of “Drag n Drop” controls with engine management:

1. The order of the engines on the Control bar can be changed
2. Engines can be added to the active Tab’s control bar by dragging them from the Measurement Configuration dialog.

Phantom Engine

Located at the bottom of the engine controls section of the Control Bar. Users can create new engines by clicking this button.  This navigates the user to the create new engine dialog. (Same as the hot key Cmd + [S] or [T])

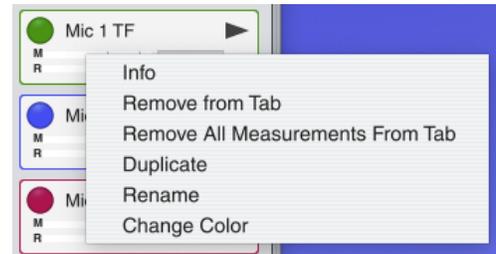


WHAT'S NEW IN SMAART 9.0

A very handy addition to the New Engine dialogs is the “Create +” button which allows a user to save a newly defined engine, and then immediately begin configuring another without closing the dialog.

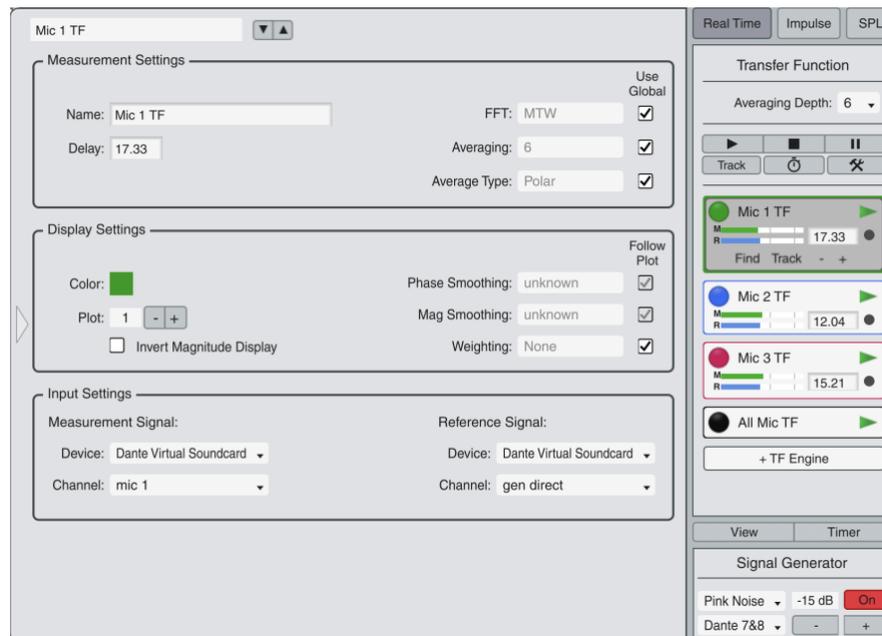
Engine Right-Click Menu

In order to reduce the navigation time to access certain engine management features, a new engine options menu can be accessed by right-clicking on a given engine. This is extremely helpful for changing engine color or for removing an engine from the active tab.



New Engine Detail

The new measurement engine detail dialog. Once opened, users can switch between the detail for active engines by clicking on the engine in the control bar. (Note that the dialog separates Measurement and Display settings)



Updated Measurement Controls

Some new/enhanced functionality has been added to the standard measurement controls in v9. These changes modify the existing operation of some key, established features.

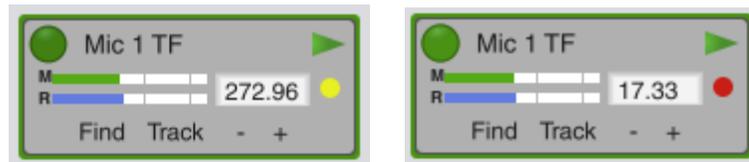
Tracker Function

As anyone who has attended a Rational Acoustics' Smart Operator Fundamentals training has probably heard numerous times – “remember to turn your tracker off.”

Setting your measurement delay using the tracker function is an extremely quick and convenient way to speed up / streamline your measurement process. Unfortunately, once you have acquired that delay

time, and assuming it is stable (issues like wind or a swaying speaker are good examples of factors that create an unstable/varying measurement delay), leaving the tracker running can only serve to frustrate you later. Particularly when making measurements to determine relative system timing, keeping your measurement delay (your timing reference) fixed is critical for comparing timing data. As you turn on and off sources you wish to time, leaving the tracker engaged will allow it to change your measurement timing – often resulting in useless measurements and foul language. Hence the constant comment in class to remember to turn off the tracker!

Now in v9, the Tracker function has been fundamentally updated to address this issue. Now when you engage the delay tracker, the tracker indicator in the engine will light up yellow as it acquires and sets the delay. Once it has found that delay and it remains stable over a few measurement cycles, the tracker will turn off automatically. If instead you wish to keep the tracker engaged, like in the case of an unstable delay time, simply double-click the tracker button and it will be locked on as before and the indicator will light up orange. The tracker can then be disengaged but clicking the tracker button on more time.



Pause

Years ago, when beta testing v7, we asked our users what they felt the default behavior should be when turning off a measurement engine. The answer came back pretty consistently that the data buffer should be kept (paused) and trace on the plot should be hidden. Then to view the data of that paused engine, the user simply clicked the show/hide button in the engine (currently displaying an “X” on it) to display the paused data.

Now in v9, we have added a “pause all” button  to the group measurement controls in the control bar. Pressing this button stops all currently running measurement engines but does not hide their traces.

New Group Measurement Controls

The group measurement controls for v9: (left to right, upper to lower)

Run All, Stop All, Pause All, Track All, Timer Function, View Measurement Config



New TF Averaging Depth Setting

The averaging settings for Transfer Function measurements have become an interesting list of choices over the past two versions. The selections available offer a mixture of FIFO and Integrating averaging choices. (2 FIFO, 4 FIFO, 8 FIFO, 16 FIFO, 1 Sec – 10 Sec, Inf, Fast and Slow), however at this point, all of

those names are misleading. The FIFO averages are based on our standard 24 measurements/sec operation rate instead of simple end-to-end or set-overlap FFT's. This means that Smaart's FIFO measurement settings do not in any way correspond with that of other dual-channel FFT system's averaging. 16 FIFO in Smaart is more averaging than 2 FIFO in Smaart, but there is little correlation between 16 FIFO in Smaart with 16 FIFO in another system.

Moreover, with changes we have made to enhance our integrating averagers (1 Sec – 10 sec), the 1-10 number is basically just an indicator of how much (8 sec is more than 2 sec averaging) but the actual integration time varies by frequency for a given setting.

The issue here then is that the names of the averaging depths (beside Infinite, Fast and Slow which do have established definitions) carry very little rigorous meaning. To be honest, the most important issue is just the depth of averaging (how much) and not the method. We have always arranged our averaging selection lists primarily by effective depth, from less to more.

Now with v9, we have jettisoned the old averaging depth names, and just settled on a simple 1-10 (+ infinite) scale for the TF averaging setting where 1 is the least amount of averaging, and 10 and infinite are the most. Want to increase stability and reject noise better? Increase the averaging depth. Want more responsiveness? Lower it.

IR and SPL Mode Updates

A great majority of the update work for the IR and SPL modes was actually completed and released as part of the v8.4 and v8.5 upgrade cycles. These changes are documented in the v8.5 Smaart User Manual released earlier this year.

In the v9 upgrade, the most significant enhancements to these two modes in the update of their GUI's to match the v9 style and in-plot controls.

Enhance – Adding New Capabilities

Arguably, this section represents the primary reason for the v9 upgrade cycle. Much of what we have spent our time working on in the v9 upgrade cycle has ultimately been towards making enhancing the Smaart platform possible. There are many features on our (always expanding) wish list of cool stuff we/you want to do with Smaart that require significant reworks to the code to make them feasible. The features that are covered in this section represent only the beginning of what is intended for v9 in the coming months and years. And quite honestly, if the past 27 years is any indication, there will probably be just as many that we don't know we intend . . . yet.

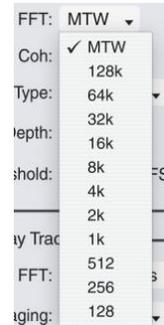
Enhanced Data Structure

In earlier versions of Smaart, there were features we did not implement because they required working with FFT's and correspondingly large data sets. As is the way of computers, what once was prohibitively large has now become reasonable and manageable.

A major undertaking in Smaart v9 has been to overhaul, and in most cases significantly increase the measurement data we are acquiring, working with and storing.

Larger Live FFT's Available in RT mode

At one time, the thought of running a single 16k FFT as part of a real-time measurement was quite daunting. For the last ten years, we had found that a 32k FFT was useful upper size limit. Now in v9 we have increased that upper limit for real-time (24 measurements per second) application in Smaart to quadruple that at 128 k. This is particularly interesting as we begin regularly using higher sample rate data – to effectively keep the same FFT Time Constants when you double (48 to 96 kHz) the sample rate, you need to double the FFT size (32K to 64K).



Make no mistake, these new increased FFT sizes (64K and 128K) can pose a significant challenge our CPU's and GPU's – particularly in older computers. But this increase does open up some interesting development possibilities.

Single FFT and MTW Always Calculated

An issue we ran into during v9 development was that, when looking at adding some new measurement capabilities, there are unique benefits to using MTW FFT's as well as to using single FFT's for our transfer function measurements. There are many reasons why we have standardized on MTW for our measurement default in Smaart. However, there are some extremely interesting measurement capabilities we wish to add into v9 that specifically require the use of a single FFT data.

MTW or single FFT? We believed ourselves to be at development fork in the road.

So we took it.

For v9 TF engines, we decided to always calculate BOTH an MTW and a single FFT version of our TF measurements. In many ways, we have been close to doing that already in v8 when we run the LiveIR and a MTW transfer function simultaneously. While this decision means we are adding some additional calculation and data management costs for each engine we run, our modern computers can definitely handle it, and they are only getting more capable each year. In return, by creating a full set of both single-FFT and MTW-based measurement data, this creates a much more complete data set to work from and opens up some very interesting feature possibilities.

Imagine if we decided not to keep the Spectrograph data buffered because it ate up too much RAM (over 64Mb!!!).

We do still have some issues to solve regarding how we select and control which data set to display. Currently, for each engine and each captured v9 reference trace file (.trf), the data displayed will match the current global FFT setting specified in the Measurement Settings dialog. Both sets are there, it is just down to refining how we choose which one to use. For now, operation will feel much the same as things worked in previous versions.

Use MTW Coh Option

One of the distinct advantages of using MTW for your TF measurement is that the COH trace is much more sensitive to reverberant energy and faulty measurement timing at higher frequencies due to the

WHAT'S NEW IN SMART 9.0

progressively shorter FFT time constants used in the HF measurements. This in turn is very useful information when using COH to provide an indication of proper measurement delay setting as well as an indication of system direct to reverberant ratios and intelligibility.

By calculating both MTW and single FFT TF, regardless of what FFT type you have selected in TF Measurement settings, v9 provides an option to always show the MTW COH trace.

MTW+ Replaces old MTW

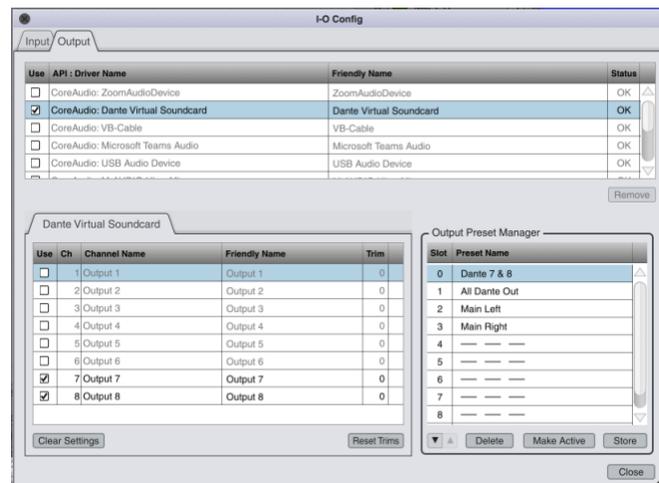
Another part of v9 development that overlapped with the last releases of v8, the enhanced MTW transfer function (MTW+) was created to remove any data time gaps (gaps between subsequent FFT's) in the HF and VHF. Standard Smart real-time operation runs the FFTs used in our measurements 24 times a second – or about every 42 milliseconds. In the high and very high frequencies, the FFT's being used in the MTW TF are shorter than that 42 ms in length, and so the HF and VHF measurements had some time gaps – they weren't effectively continuous. These gaps do not pose a problem with the accuracy of the response measurement, however it does create a situation where it could miss short transient (distortion) events.

MTW+ runs the FFT's in the HF so that they are effectively continuous. The MTW+ routine developed during the last year of v8 effectively replaced the old MTW in v9.

Expanded Output Capabilities

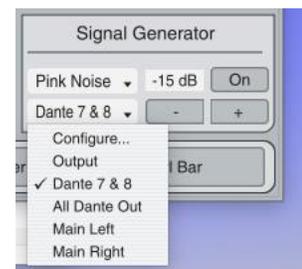
Previous versions of Smart have been limited to a single **mono** output signal sent on up to two (2) output channels. When playing stereo .wav files from Smart's generator, only the Left side of the signal would be played.

In v9, this limitation has been removed. Smart's generator output can now be set to send signals out any and available output channels. Moreover, as Smart's generator expands to include stereo and multi-channel playback and noise sources, so will the output routing.



Output Presets

Pre-set output configurations can be configured and stored in v9's I-O Output Config. These output pre-sets can then be selected from Smart signal generator control area at the bottom of the control bar to switch between your pre-set output drive configurations.



Timer (Macro) Function

SmaartLive v5 had a timed measurement function that has always been earmarked to make its way back into the current Smaart products. Version 9 marks its re-introduction.

The basic operation of the Timer function is that when you activate the timed measurement, for all actively running measurements Smaart:

- sets its average type to “Infinite”
- flushes out the averaging buffers
- averages new data for the specified length of time (specified in seconds)
- captures the data
- resets the average type back to its previous setting

This little macro function represents a quick way of collecting measurement data. In some cases, most notably some cinema tuning processes, it was used to do thirty-second-long walking averages with spectrum measurements – basically walking the microphone around through the listening area of a theatre while the sound system is energized with pink noise.



With the original Timer function in v5, we eventually added options to include a countdown before starting the measurement (so the engineer could activate the function and still have time to get into place for the measurement), and the ability to turn on and off the generator (so the engineer is aware of when the measurement has started and has ended).

One of the major goals for v9 is to introduce user definable macros into the Smaart toolbox. It has been an often-requested feature for a user to create macros to automate some configuration and measurement processes that they perform repeatedly. This Timer function represents our first foray into this user definable macro capability.

The Timer macro is relatively simple with a few basic options to modify it. The configuration of the timer is done via the timer configuration pop-up in the control bar. Besides the basic functionality detailed above, additional steps can be added to the process.

Those optional steps are (in order of execution in the macro) - for each active running measurement engine:

1. Count Down [specified in seconds]
2. Turn Generator On
3. Set Measurement Delay (uses the find function – TF measurement only)
4. Track Delay (locks tracker on – TF measurement only)
5. Perform Timed Average (see above process)
6. Turn Generator Off
7. Repeat [X] times

Parting Reminders

For questions, comments, bug reports, feature requests, or anything Smaart related, contact:

SUPPORT@RATIONALACOUSTICS.COM

Also, remember the help files! Thank you.