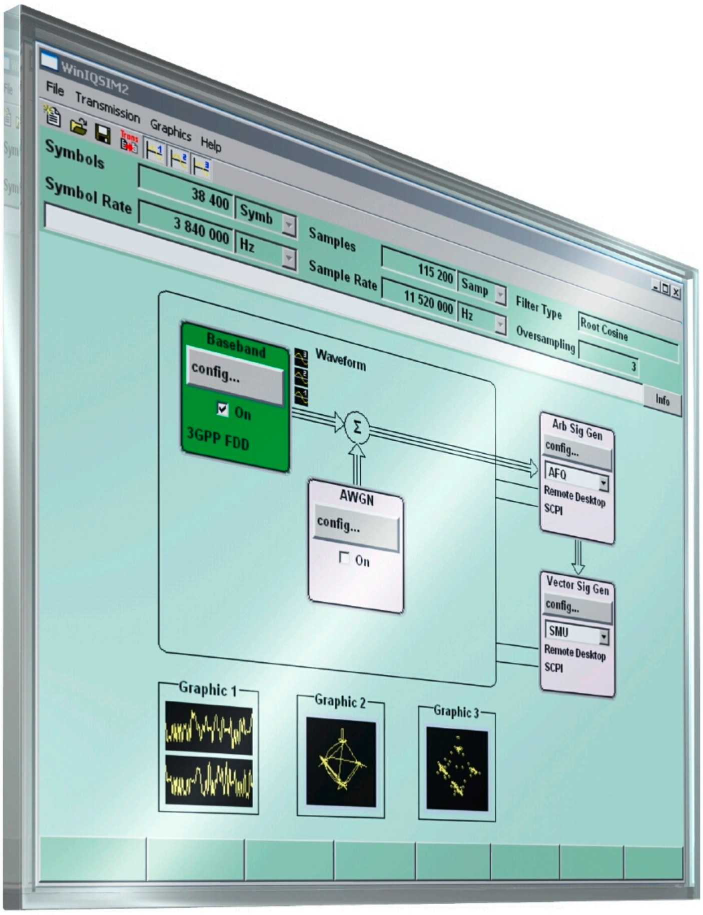
R&S®WinIQSIM2
Signal Generation Software
Software Manual



Symbols: 38 400 Symb

Symbol Rate: 3 840 000 Hz

Samples: 115 200 Samp

Sample Rate: 11 520 000 Hz

Filter Type: Root Cosine

Oversampling: 3

1008.0858.42 — 09

Test & Measurement

Software Manual

ROHDE&SCHWARZ

This document describes the R&S®WinIQSIM2, stock no. 1008.0858.42 and its options.

- R&S®WinIQSIM-K261
- R&S®WinIQSIM-K262

The Signal Generation Software makes use of several valuable open source software packages. The most important of them together with their corresponding open source license and the verbatim license texts are provided on the user documentation CD-ROM (included in delivery).
The OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/) includes cryptographic software written by Eric Young (eay@cryptsoft.com) and software written by Tim Hudson (tjh@cryptsoft.com).
Rohde & Schwarz would like to thank the open source community for their valuable contribution to embedded computing.

© 2012 Rohde & Schwarz GmbH & Co. KG
Muehldorfstr. 15, 81671 Munich, Germany
Phone: +49 89 41 29 - 0
Fax: +49 89 41 29 12 164
E-mail: info@rohde-schwarz.com
Internet: http://www.rohde-schwarz.com
Printed in Germany – Subject to change – Data without tolerance limits is not binding.

The following abbreviations are used throughout this manual: R&S®WinIQSIM2 is abbreviated as R&S WinIQSIM2, R&S®AMU is abbreviated as R&S AMU, R&S®AFQ100A is abbreviated as R&S AFQ100A, R&S®AFQ100B is abbreviated as R&S AFQ100B,R&S®SMATE is abbreviated as R&S SMATE, R&S®SMBV is abbreviated as R&S SMBV, R&S®SMJ is abbreviated as R&S SMJ, R&S®SMU is abbreviated as R&S SMU, R&S®EX-IQ-BOX is abbreviated as R&S EX-IQ-BOX.

Basic Safety Instructions

Always read through and comply with the following safety instructions!

All plants and locations of the Rohde & Schwarz group of companies make every effort to keep the safety standards of our products up to date and to offer our customers the highest possible degree of safety. Our products and the auxiliary equipment they require are designed, built and tested in accordance with the safety standards that apply in each case. Compliance with these standards is continuously monitored by our quality assurance system. The product described here has been designed, built and tested in accordance with the attached EC Certificate of Conformity and has left the manufacturer’s plant in a condition fully complying with safety standards. To maintain this condition and to ensure safe operation, you must observe all instructions and warnings provided in this manual. If you have any questions regarding these safety instructions, the Rohde & Schwarz group of companies will be happy to answer them.

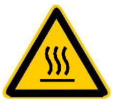
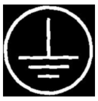
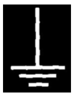
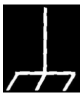
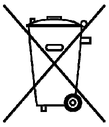
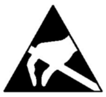
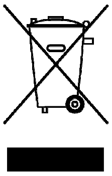
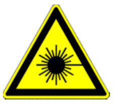
Furthermore, it is your responsibility to use the product in an appropriate manner. This product is designed for use solely in industrial and laboratory environments or, if expressly permitted, also in the field and must not be used in any way that may cause personal injury or property damage. You are responsible if the product is used for any purpose other than its designated purpose or in disregard of the manufacturer's instructions. The manufacturer shall assume no responsibility for such use of the product.

The product is used for its designated purpose if it is used in accordance with its product documentation and within its performance limits (see data sheet, documentation, the following safety instructions). Using the product requires technical skills and, in some cases, a basic knowledge of English. It is therefore essential that only skilled and specialized staff or thoroughly trained personnel with the required skills be allowed to use the product. If personal safety gear is required for using Rohde & Schwarz products, this will be indicated at the appropriate place in the product documentation. Keep the basic safety instructions and the product documentation in a safe place and pass them on to the subsequent users.

Observing the safety instructions will help prevent personal injury or damage of any kind caused by dangerous situations. Therefore, carefully read through and adhere to the following safety instructions before and when using the product. It is also absolutely essential to observe the additional safety instructions on personal safety, for example, that appear in relevant parts of the product documentation. In these safety instructions, the word "product" refers to all merchandise sold and distributed by the Rohde & Schwarz group of companies, including instruments, systems and all accessories. For product-specific information, see the data sheet and the product documentation.

Symbols and safety labels

Symbol	Meaning	Symbol	Meaning
⚠	Notice, general danger location Observe product documentation	| ○	ON/OFF supply voltage
⚠ 18 kg	Caution when handling heavy equipment	⏻	Standby indication
⚡	Danger of electric shock	⎓	Direct current (DC)

Symbol	Meaning	Symbol	Meaning
	Warning! Hot surface	~	Alternating current (AC)
	Protective conductor terminal	≂	Direct/alternating current (DC/AC)
	Ground	⧈	Device fully protected by double (reinforced) insulation
	Ground terminal		EU labeling for batteries and accumulators For additional information, see section "Waste disposal/Environmental protection", item 1.
	Be careful when handling electrostatic sensitive devices		EU labeling for separate collection of electrical and electronic devices For additonal information, see section "Waste disposal/Environmental protection", item 2.
	Warning! Laser radiation For additional information, see section "Operation", item 7.		

Signal words and their meaning

The following signal words are used in the product documentation in order to warn the reader about risks and dangers.

⚠ DANGER Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE Indicates the possibility of incorrect operation which can result in damage to the product.
In the product documentation, the word ATTENTION is used synonymously.

These signal words are in accordance with the standard definition for civil applications in the European Economic Area. Definitions that deviate from the standard definition may also exist in other economic areas or military applications. It is therefore essential to make sure that the signal words described here are always used only in connection with the related product documentation and the related product. The use of signal words in connection with unrelated products or documentation can result in misinterpretation and in personal injury or material damage.

Operating states and operating positions

The product may be operated only under the operating conditions and in the positions specified by the manufacturer, without the product's ventilation being obstructed. If the manufacturer's specifications are not observed, this can result in electric shock, fire and/or serious personal injury or death. Applicable local or national safety regulations and rules for the prevention of accidents must be observed in all work performed.

1. Unless otherwise specified, the following requirements apply to Rohde & Schwarz products: predefined operating position is always with the housing floor facing down, IP protection 2X, use only indoors, max. operating altitude 2000 m above sea level, max. transport altitude 4500 m above sea level. A tolerance of ±10 % shall apply to the nominal voltage and ±5 % to the nominal frequency, overvoltage category 2, pollution severity 2.
2. Do not place the product on surfaces, vehicles, cabinets or tables that for reasons of weight or stability are unsuitable for this purpose. Always follow the manufacturer's installation instructions when installing the product and fastening it to objects or structures (e.g. walls and shelves). An installation that is not carried out as described in the product documentation could result in personal injury or even death.
3. Do not place the product on heat-generating devices such as radiators or fan heaters. The ambient temperature must not exceed the maximum temperature specified in the product documentation or in the data sheet. Product overheating can cause electric shock, fire and/or serious personal injury or even death.

Electrical safety

If the information on electrical safety is not observed either at all or to the extent necessary, electric shock, fire and/or serious personal injury or death may occur.

1. Prior to switching on the product, always ensure that the nominal voltage setting on the product matches the nominal voltage of the AC supply network. If a different voltage is to be set, the power fuse of the product may have to be changed accordingly.
2. In the case of products of safety class I with movable power cord and connector, operation is permitted only on sockets with a protective conductor contact and protective conductor.
3. Intentionally breaking the protective conductor either in the feed line or in the product itself is not permitted. Doing so can result in the danger of an electric shock from the product. If extension cords or connector strips are implemented, they must be checked on a regular basis to ensure that they are safe to use.
4. If there is no power switch for disconnecting the product from the AC supply network, or if the power switch is not suitable for this purpose, use the plug of the connecting cable to disconnect the product from the AC supply network. In such cases, always ensure that the power plug is easily reachable and accessible at all times. For example, if the power plug is the disconnecting device, the length of the connecting cable must not exceed 3 m. Functional or electronic switches are not suitable for providing disconnection from the AC supply network. If products without power switches are integrated into racks or systems, the disconnecting device must be provided at the system level.
5. Never use the product if the power cable is damaged. Check the power cables on a regular basis to ensure that they are in proper operating condition. By taking appropriate safety measures and carefully laying the power cable, ensure that the cable cannot be damaged and that no one can be hurt by, for example, tripping over the cable or suffering an electric shock.

6. The product may be operated only from TN/TT supply networks fuse-protected with max. 16 A (higher fuse only after consulting with the Rohde & Schwarz group of companies).
7. Do not insert the plug into sockets that are dusty or dirty. Insert the plug firmly and all the way into the socket provided for this purpose. Otherwise, sparks that result in fire and/or injuries may occur.
8. Do not overload any sockets, extension cords or connector strips; doing so can cause fire or electric shocks.
9. For measurements in circuits with voltages $V_{rms} > 30$ V, suitable measures (e.g. appropriate measuring equipment, fuse protection, current limiting, electrical separation, insulation) should be taken to avoid any hazards.
10. Ensure that the connections with information technology equipment, e.g. PCs or other industrial computers, comply with the IEC60950-1/EN60950-1 or IEC61010-1/EN 61010-1 standards that apply in each case.
11. Unless expressly permitted, never remove the cover or any part of the housing while the product is in operation. Doing so will expose circuits and components and can lead to injuries, fire or damage to the product.
12. If a product is to be permanently installed, the connection between the protective conductor terminal on site and the product's protective conductor must be made first before any other connection is made. The product may be installed and connected only by a licensed electrician.
13. For permanently installed equipment without built-in fuses, circuit breakers or similar protective devices, the supply circuit must be fuse-protected in such a way that anyone who has access to the product, as well as the product itself, is adequately protected from injury or damage.
14. Use suitable overvoltage protection to ensure that no overvoltage (such as that caused by a bolt of lightning) can reach the product. Otherwise, the person operating the product will be exposed to the danger of an electric shock.
15. Any object that is not designed to be placed in the openings of the housing must not be used for this purpose. Doing so can cause short circuits inside the product and/or electric shocks, fire or injuries.
16. Unless specified otherwise, products are not liquid-proof (see also section "Operating states and operating positions", item 1). Therefore, the equipment must be protected against penetration by liquids. If the necessary precautions are not taken, the user may suffer electric shock or the product itself may be damaged, which can also lead to personal injury.
17. Never use the product under conditions in which condensation has formed or can form in or on the product, e.g. if the product has been moved from a cold to a warm environment. Penetration by water increases the risk of electric shock.
18. Prior to cleaning the product, disconnect it completely from the power supply (e.g. AC supply network or battery). Use a soft, non-linting cloth to clean the product. Never use chemical cleaning agents such as alcohol, acetone or diluents for cellulose lacquers.

Operation

1. Operating the products requires special training and intense concentration. Make sure that persons who use the products are physically, mentally and emotionally fit enough to do so; otherwise, injuries or material damage may occur. It is the responsibility of the employer/operator to select suitable personnel for operating the products.

2. Before you move or transport the product, read and observe the section titled "Transport".
3. As with all industrially manufactured goods, the use of substances that induce an allergic reaction (allergens) such as nickel cannot be generally excluded. If you develop an allergic reaction (such as a skin rash, frequent sneezing, red eyes or respiratory difficulties) when using a Rohde & Schwarz product, consult a physician immediately to determine the cause and to prevent health problems or stress.
4. Before you start processing the product mechanically and/or thermally, or before you take it apart, be sure to read and pay special attention to the section titled "Waste disposal/Environmental protection", item 1.
5. Depending on the function, certain products such as RF radio equipment can produce an elevated level of electromagnetic radiation. Considering that unborn babies require increased protection, pregnant women must be protected by appropriate measures. Persons with pacemakers may also be exposed to risks from electromagnetic radiation. The employer/operator must evaluate workplaces where there is a special risk of exposure to radiation and, if necessary, take measures to avert the potential danger.
6. Should a fire occur, the product may release hazardous substances (gases, fluids, etc.) that can cause health problems. Therefore, suitable measures must be taken, e.g. protective masks and protective clothing must be worn.
7. Laser products are given warning labels that are standardized according to their laser class. Lasers can cause biological harm due to the properties of their radiation and due to their extremely concentrated electromagnetic power. If a laser product (e.g. a CD/DVD drive) is integrated into a Rohde & Schwarz product, absolutely no other settings or functions may be used as described in the product documentation. The objective is to prevent personal injury (e.g. due to laser beams).
8. EMC classes (in line with CISPR 11)
   Class A: Equipment suitable for use in all environments except residential environments and environments that are directly connected to a low-voltage supply network that supplies residential buildings.
   Class B: Equipment suitable for use in residential environments and environments that are directly connected to a low-voltage supply network that supplies residential buildings.

Repair and service

1. The product may be opened only by authorized, specially trained personnel. Before any work is performed on the product or before the product is opened, it must be disconnected from the AC supply network. Otherwise, personnel will be exposed to the risk of an electric shock.
2. Adjustments, replacement of parts, maintenance and repair may be performed only by electrical experts authorized by Rohde & Schwarz. Only original parts may be used for replacing parts relevant to safety (e.g. power switches, power transformers, fuses). A safety test must always be performed after parts relevant to safety have been replaced (visual inspection, protective conductor test, insulation resistance measurement, leakage current measurement, functional test). This helps ensure the continued safety of the product.

Batteries and rechargeable batteries/cells

If the information regarding batteries and rechargeable batteries/cells is not observed either at all or to the extent necessary, product users may be exposed to the risk of explosions, fire and/or serious personal injury, and, in some cases, death. Batteries and rechargeable batteries with alkaline electrolytes (e.g. lithium cells) must be handled in accordance with the EN 62133 standard.

1. Cells must not be taken apart or crushed.
2. Cells or batteries must not be exposed to heat or fire. Storage in direct sunlight must be avoided. Keep cells and batteries clean and dry. Clean soiled connectors using a dry, clean cloth.
3. Cells or batteries must not be short-circuited. Cells or batteries must not be stored in a box or in a drawer where they can short-circuit each other, or where they can be short-circuited by other conductive materials. Cells and batteries must not be removed from their original packaging until they are ready to be used.
4. Cells and batteries must not be exposed to any mechanical shocks that are stronger than permitted.
5. If a cell develops a leak, the fluid must not be allowed to come into contact with the skin or eyes. If contact occurs, wash the affected area with plenty of water and seek medical aid.
6. Improperly replacing or charging cells or batteries that contain alkaline electrolytes (e.g. lithium cells) can cause explosions. Replace cells or batteries only with the matching Rohde & Schwarz type (see parts list) in order to ensure the safety of the product.
7. Cells and batteries must be recycled and kept separate from residual waste. Rechargeable batteries and normal batteries that contain lead, mercury or cadmium are hazardous waste. Observe the national regulations regarding waste disposal and recycling.

Transport

1. The product may be very heavy. Therefore, the product must be handled with care. In some cases, the user may require a suitable means of lifting or moving the product (e.g. with a lift-truck) to avoid back or other physical injuries.
2. Handles on the products are designed exclusively to enable personnel to transport the product. It is therefore not permissible to use handles to fasten the product to or on transport equipment such as cranes, fork lifts, wagons, etc. The user is responsible for securely fastening the products to or on the means of transport or lifting. Observe the safety regulations of the manufacturer of the means of transport or lifting. Noncompliance can result in personal injury or material damage.
3. If you use the product in a vehicle, it is the sole responsibility of the driver to drive the vehicle safely and properly. The manufacturer assumes no responsibility for accidents or collisions. Never use the product in a moving vehicle if doing so could distract the driver of the vehicle. Adequately secure the product in the vehicle to prevent injuries or other damage in the event of an accident.

Waste disposal/Environmental protection

1. Specially marked equipment has a battery or accumulator that must not be disposed of with unsorted municipal waste, but must be collected separately. It may only be disposed of at a suitable collection point or via a Rohde & Schwarz customer service center.

2. Waste electrical and electronic equipment must not be disposed of with unsorted municipal waste, but must be collected separately.
   Rohde & Schwarz GmbH & Co. KG has developed a disposal concept and takes full responsibility for take-back obligations and disposal obligations for manufacturers within the EU. Contact your Rohde & Schwarz customer service center for environmentally responsible disposal of the product.
3. If products or their components are mechanically and/or thermally processed in a manner that goes beyond their intended use, hazardous substances (heavy-metal dust such as lead, beryllium, nickel) may be released. For this reason, the product may only be disassembled by specially trained personnel. Improper disassembly may be hazardous to your health. National waste disposal regulations must be observed.
4. If handling the product releases hazardous substances or fuels that must be disposed of in a special way, e.g. coolants or engine oils that must be replenished regularly, the safety instructions of the manufacturer of the hazardous substances or fuels and the applicable regional waste disposal regulations must be observed. Also observe the relevant safety instructions in the product documentation. The improper disposal of hazardous substances or fuels can cause health problems and lead to environmental damage.

For additional information about environmental protection, visit the Rohde & Schwarz website.

Instrucciones de seguridad elementales

¡Es imprescindible leer y cumplir las siguientes instrucciones e informaciones de seguridad!

El principio del grupo de empresas Rohde & Schwarz consiste en tener nuestros productos siempre al día con los estándares de seguridad y de ofrecer a nuestros clientes el máximo grado de seguridad. Nuestros productos y todos los equipos adicionales son siempre fabricados y examinados según las normas de seguridad vigentes. Nuestro sistema de garantía de calidad controla constantemente que sean cumplidas estas normas. El presente producto ha sido fabricado y examinado según el certificado de conformidad adjunto de la UE y ha salido de nuestra planta en estado impecable según los estándares técnicos de seguridad. Para poder preservar este estado y garantizar un funcionamiento libre de peligros, el usuario deberá atenerse a todas las indicaciones, informaciones de seguridad y notas de alerta. El grupo de empresas Rohde & Schwarz está siempre a su disposición en caso de que tengan preguntas referentes a estas informaciones de seguridad.

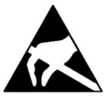
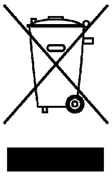
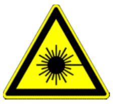
Además queda en la responsabilidad del usuario utilizar el producto en la forma debida. Este producto está destinado exclusivamente al uso en la industria y el laboratorio o, si ha sido expresamente autorizado, para aplicaciones de campo y de ninguna manera deberá ser utilizado de modo que alguna persona/cosa pueda sufrir daño. El uso del producto fuera de sus fines definidos o sin tener en cuenta las instrucciones del fabricante queda en la responsabilidad del usuario. El fabricante no se hace en ninguna forma responsable de consecuencias a causa del mal uso del producto.

Se parte del uso correcto del producto para los fines definidos si el producto es utilizado conforme a las indicaciones de la correspondiente documentación del producto y dentro del margen de rendimiento definido (ver hoja de datos, documentación, informaciones de seguridad que siguen). El uso del producto hace necesarios conocimientos técnicos y ciertos conocimientos del idioma inglés. Por eso se debe tener en cuenta que el producto solo pueda ser operado por personal especializado o personas instruidas en profundidad con las capacidades correspondientes. Si fuera necesaria indumentaria de seguridad para el uso de productos de Rohde & Schwarz, encontraría la información debida en la documentación del producto en el capítulo correspondiente. Guarde bien las informaciones de seguridad elementales, así como la documentación del producto, y entréguelas a usuarios posteriores.

Tener en cuenta las informaciones de seguridad sirve para evitar en lo posible lesiones o daños por peligros de toda clase. Por eso es imprescindible leer detalladamente y comprender por completo las siguientes informaciones de seguridad antes de usar el producto, y respetarlas durante el uso del producto. Deberán tenerse en cuenta todas las demás informaciones de seguridad, como p. ej. las referentes a la protección de personas, que encontrarán en el capítulo correspondiente de la documentación del producto y que también son de obligado cumplimiento. En las presentes informaciones de seguridad se recogen todos los objetos que distribuye el grupo de empresas Rohde & Schwarz bajo la denominación de "producto", entre ellos también aparatos, instalaciones así como toda clase de accesorios. Los datos específicos del producto figuran en la hoja de datos y en la documentación del producto.

Símbolos y definiciones de seguridad

Símbolo	Significado	Símbolo	Significado
⚠	Aviso: punto de peligro general Observar la documentación del producto	| ○	Tensión de alimentación de PUESTA EN MARCHA / PARADA
⚠ 18 kg	Atención en el manejo de dispositivos de peso elevado	⏻	Indicación de estado de espera (standby)
⚡	Peligro de choque eléctrico	⎓	Corriente continua (DC)
♨	Advertencia: superficie caliente	∼	Corriente alterna (AC)
⏚	Conexión a conductor de protección	≂	Corriente continua / Corriente alterna (DC/AC)
⏚	Conexión a tierra	⧈	El aparato está protegido en su totalidad por un aislamiento doble (reforzado)
⊥	Conexión a masa	☒	Distintivo de la UE para baterías y acumuladores Más información en la sección "Eliminación/protección del medio ambiente", punto 1.

Símbolo	Significado	Símbolo	Significado
	Aviso: Cuidado en el manejo de dispositivos sensibles a la electrostática (ESD)		Distintivo de la UE para la eliminación por separado de dispositivos eléctricos y electrónicos Más información en la sección "Eliminación/protección del medio ambiente", punto 2.
	Advertencia: rayo láser Más información en la sección "Funcionamiento", punto 7.		

Palabras de señal y su significado

En la documentación del producto se utilizan las siguientes palabras de señal con el fin de advertir contra riesgos y peligros.

⚠ PELIGRO	PELIGRO identifica un peligro inminente con riesgo elevado que provocará muerte o lesiones graves si no se evita.
⚠ ADVERTENCIA	ADVERTENCIA identifica un posible peligro con riesgo medio de provocar muerte o lesiones (graves) si no se evita.
⚠ ATENCIÓN	ATENCIÓN identifica un peligro con riesgo reducido de provocar lesiones leves o moderadas si no se evita.
AVISO	AVISO indica la posibilidad de utilizar mal el producto y, como consecuencia, dañarlo. En la documentación del producto se emplea de forma sinónima el término CUIDADO.

Las palabras de señal corresponden a la definición habitual para aplicaciones civiles en el área económica europea. Pueden existir definiciones diferentes a esta definición en otras áreas económicas o en aplicaciones militares. Por eso se deberá tener en cuenta que las palabras de señal aquí descritas sean utilizadas siempre solamente en combinación con la correspondiente documentación del producto y solamente en combinación con el producto correspondiente. La utilización de las palabras de señal en combinación con productos o documentaciones que no les correspondan puede llevar a interpretaciones equivocadas y tener por consecuencia daños en personas u objetos.

Estados operativos y posiciones de funcionamiento

El producto solamente debe ser utilizado según lo indicado por el fabricante respecto a los estados operativos y posiciones de funcionamiento sin que se obstruya la ventilación. Si no se siguen las indicaciones del fabricante, pueden producirse choques eléctricos, incendios y/o lesiones graves con posible consecuencia de muerte. En todos los trabajos deberán ser tenidas en cuenta las normas nacionales y locales de seguridad del trabajo y de prevención de accidentes.

1. Si no se convino de otra manera, es para los productos Rohde & Schwarz válido lo que sigue: como posición de funcionamiento se define por principio la posición con el suelo de la caja para abajo, modo de protección IP 2X, uso solamente en estancias interiores, utilización hasta 2000 m sobre el nivel del mar, transporte hasta 4500 m sobre el nivel del mar. Se aplicará una tolerancia de ±10 % sobre el voltaje nominal y de ±5 % sobre la frecuencia nominal. Categoría de sobrecarga eléctrica 2, índice de suciedad 2.
2. No sitúe el producto encima de superficies, vehículos, estantes o mesas, que por sus características de peso o de estabilidad no sean aptos para él. Siga siempre las instrucciones de instalación del fabricante cuando instale y asegure el producto en objetos o estructuras (p. ej. paredes y estantes). Si se realiza la instalación de modo distinto al indicado en la documentación del producto, se pueden causar lesiones o, en determinadas circunstancias, incluso la muerte.
3. No ponga el producto sobre aparatos que generen calor (p. ej. radiadores o calefactores). La temperatura ambiente no debe superar la temperatura máxima especificada en la documentación del producto o en la hoja de datos. En caso de sobrecalentamiento del producto, pueden producirse choques eléctricos, incendios y/o lesiones graves con posible consecuencia de muerte.

Seguridad eléctrica

Si no se siguen (o se siguen de modo insuficiente) las indicaciones del fabricante en cuanto a seguridad eléctrica, pueden producirse choques eléctricos, incendios y/o lesiones graves con posible consecuencia de muerte.

1. Antes de la puesta en marcha del producto se deberá comprobar siempre que la tensión preseleccionada en el producto coincida con la de la red de alimentación eléctrica. Si es necesario modificar el ajuste de tensión, también se deberán cambiar en caso dado los fusibles correspondientes del producto.
2. Los productos de la clase de protección I con alimentación móvil y enchufe individual solamente podrán enchufarse a tomas de corriente con contacto de seguridad y con conductor de protección conectado.
3. Queda prohibida la interrupción intencionada del conductor de protección, tanto en la toma de corriente como en el mismo producto. La interrupción puede tener como consecuencia el riesgo de que el producto sea fuente de choques eléctricos. Si se utilizan cables alargadores o regletas de enchufe, deberá garantizarse la realización de un examen regular de los mismos en cuanto a su estado técnico de seguridad.
4. Si el producto no está equipado con un interruptor para desconectarlo de la red, o bien si el interruptor existente no resulta apropiado para la desconexión de la red, el enchufe del cable de conexión se deberá considerar como un dispositivo de desconexión.
   El dispositivo de desconexión se debe poder alcanzar fácilmente y debe estar siempre bien accesible. Si, p. ej., el enchufe de conexión a la red es el dispositivo de desconexión, la longitud del cable de conexión no debe superar 3 m).
   Los interruptores selectores o electrónicos no son aptos para el corte de la red eléctrica. Si se integran productos sin interruptor en bastidores o instalaciones, se deberá colocar el interruptor en el nivel de la instalación.
5. No utilice nunca el producto si está dañado el cable de conexión a red. Compruebe regularmente el correcto estado de los cables de conexión a red. Asegúrese, mediante las medidas de protección y de instalación adecuadas, de que el cable de conexión a red no pueda ser dañado o de que nadie pueda ser dañado por él, p. ej. al tropezar o por un choque eléctrico.

6. Solamente está permitido el funcionamiento en redes de alimentación TN/TT aseguradas con fusibles de 16 A como máximo (utilización de fusibles de mayor amperaje solo previa consulta con el grupo de empresas Rohde & Schwarz).
7. Nunca conecte el enchufe en tomas de corriente sucias o llenas de polvo. Introduzca el enchufe por completo y fuertemente en la toma de corriente. La no observación de estas medidas puede provocar chispas, fuego y/o lesiones.
8. No sobrecargue las tomas de corriente, los cables alargadores o las regletas de enchufe ya que esto podría causar fuego o choques eléctricos.
9. En las mediciones en circuitos de corriente con una tensión U_{eff} > 30 V se deberán tomar las medidas apropiadas para impedir cualquier peligro (p. ej. medios de medición adecuados, seguros, limitación de tensión, corte protector, aislamiento etc.).
10. Para la conexión con dispositivos informáticos como un PC o un ordenador industrial, debe comprobarse que éstos cumplan los estándares IEC60950-1/EN60950-1 o IEC61010-1/EN 61010-1 válidos en cada caso.
11. A menos que esté permitido expresamente, no retire nunca la tapa ni componentes de la carcasa mientras el producto esté en servicio. Esto pone a descubierto los cables y componentes eléctricos y puede causar lesiones, fuego o daños en el producto.
12. Si un producto se instala en un lugar fijo, se deberá primero conectar el conductor de protección fijo con el conductor de protección del producto antes de hacer cualquier otra conexión. La instalación y la conexión deberán ser efectuadas por un electricista especializado.
13. En el caso de dispositivos fijos que no estén provistos de fusibles, interruptor automático ni otros mecanismos de seguridad similares, el circuito de alimentación debe estar protegido de modo que todas las personas que puedan acceder al producto, así como el producto mismo, estén a salvo de posibles daños.
14. Todo producto debe estar protegido contra sobretensión (debida p. ej. a una caída del rayo) mediante los correspondientes sistemas de protección. Si no, el personal que lo utilice quedará expuesto al peligro de choque eléctrico.
15. No debe introducirse en los orificios de la caja del aparato ningún objeto que no esté destinado a ello. Esto puede producir cortocircuitos en el producto y/o puede causar choques eléctricos, fuego o lesiones.
16. Salvo indicación contraria, los productos no están impermeabilizados (ver también el capítulo "Estados operativos y posiciones de funcionamiento", punto 1). Por eso es necesario tomar las medidas necesarias para evitar la entrada de líquidos. En caso contrario, existe peligro de choque eléctrico para el usuario o de daños en el producto, que también pueden redundar en peligro para las personas.
17. No utilice el producto en condiciones en las que pueda producirse o ya se hayan producido condensaciones sobre el producto o en el interior de éste, como p. ej. al desplazarlo de un lugar frío a otro caliente. La entrada de agua aumenta el riesgo de choque eléctrico.
18. Antes de la limpieza, desconecte por completo el producto de la alimentación de tensión (p. ej. red de alimentación o batería). Realice la limpieza de los aparatos con un paño suave, que no se deshilache. No utilice bajo ningún concepto productos de limpieza químicos como alcohol, acetona o diluyentes para lacas nitrocelulósicas.

Funcionamiento

1. El uso del producto requiere instrucciones especiales y una alta concentración durante el manejo. Debe asegurarse que las personas que manejen el producto estén a la altura de los requerimientos necesarios en cuanto a aptitudes físicas, psíquicas y emocionales, ya que de otra manera no se pueden excluir lesiones o daños de objetos. El empresario u operador es responsable de seleccionar el personal usuario apto para el manejo del producto.
2. Antes de desplazar o transportar el producto, lea y tenga en cuenta el capítulo "Transporte".
3. Como con todo producto de fabricación industrial no puede quedar excluida en general la posibilidad de que se produzcan alergias provocadas por algunos materiales empleados —los llamados alérgenos (p. ej. el níquel)—. Si durante el manejo de productos Rohde & Schwarz se producen reacciones alérgicas, como p. ej. irritaciones cutáneas, estornudos continuos, enrojecimiento de la conjuntiva o dificultades respiratorias, debe avisarse inmediatamente a un médico para investigar las causas y evitar cualquier molestia o daño a la salud.
4. Antes de la manipulación mecánica y/o térmica o el desmontaje del producto, debe tenerse en cuenta imprescindiblemente el capítulo "Eliminación/protección del medio ambiente", punto 1.
5. Ciertos productos, como p. ej. las instalaciones de radiocomunicación RF, pueden a causa de su función natural, emitir una radiación electromagnética aumentada. Deben tomarse todas las medidas necesarias para la protección de las mujeres embarazadas. También las personas con marcapasos pueden correr peligro a causa de la radiación electromagnética. El empresario/operador tiene la obligación de evaluar y señalizar las áreas de trabajo en las que exista un riesgo elevado de exposición a radiaciones.
6. Tenga en cuenta que en caso de incendio pueden desprenderse del producto sustancias tóxicas (gases, líquidos etc.) que pueden generar daños a la salud. Por eso, en caso de incendio deben usarse medidas adecuadas, como p. ej. máscaras antigás e indumentaria de protección.
7. Los productos con láser están provistos de indicaciones de advertencia normalizadas en función de la clase de láser del que se trate. Los rayos láser pueden provocar daños de tipo biológico a causa de las propiedades de su radiación y debido a su concentración extrema de potencia electromagnética. En caso de que un producto Rohde & Schwarz contenga un producto láser (p. ej. un lector de CD/DVD), no debe usarse ninguna otra configuración o función aparte de las descritas en la documentación del producto, a fin de evitar lesiones (p. ej. debidas a irradiación láser).
8. Clases CEM (según CISPR 11)
   Clase A: dispositivo apropiado para el uso en cualquier zona excepto en áreas residenciales y en aquellas zonas que se encuentran conectadas a una red de suministro de baja tensión que alimenta un edificio de viviendas.
   Clase B: dispositivo apropiado para el uso en áreas residenciales y en aquellas zonas que se encuentran conectadas a una red de suministro de baja tensión que alimenta un edificio de viviendas.

Reparación y mantenimiento

1. El producto solamente debe ser abierto por personal especializado con autorización para ello. Antes de manipular el producto o abrirlo, es obligatorio desconectarlo de la tensión de alimentación, para evitar toda posibilidad de choque eléctrico.

2. El ajuste, el cambio de partes, el mantenimiento y la reparación deberán ser efectuadas solamente por electricistas autorizados por Rohde & Schwarz. Si se reponen partes con importancia para los aspectos de seguridad (p. ej. el enchufe, los transformadores o los fusibles), solamente podrán ser sustituidos por partes originales. Después de cada cambio de partes relevantes para la seguridad deberá realizarse un control de seguridad (control a primera vista, control del conductor de protección, medición de resistencia de aislamiento, medición de la corriente de fuga, control de funcionamiento). Con esto queda garantizada la seguridad del producto.

Baterías y acumuladores o celdas

Si no se siguen (o se siguen de modo insuficiente) las indicaciones en cuanto a las baterías y acumuladores o celdas, pueden producirse explosiones, incendios y/o lesiones graves con posible consecuencia de muerte. El manejo de baterías y acumuladores con electrolitos alcalinos (p. ej. celdas de litio) debe seguir el estándar EN 62133.

1. No deben desmontarse, abrirse ni triturarse las celdas.
2. Las celdas o baterías no deben someterse a calor ni fuego. Debe evitarse el almacenamiento a la luz directa del sol. Las celdas y baterías deben mantenerse limpias y secas. Limpiar las conexiones sucias con un paño seco y limpio.
3. Las celdas o baterías no deben cortocircuitarse. Es peligroso almacenar las celdas o baterías en estuches o cajones en cuyo interior puedan cortocircuitarse por contacto recíproco o por contacto con otros materiales conductores. No deben extraerse las celdas o baterías de sus embalajes originales hasta el momento en que vayan a utilizarse.
4. Las celdas o baterías no deben someterse a impactos mecánicos fuertes indebidos.
5. En caso de falta de estanqueidad de una celda, el líquido vertido no debe entrar en contacto con la piel ni los ojos. Si se produce contacto, lavar con agua abundante la zona afectada y avisar a un médico.
6. En caso de cambio o recarga inadecuados, las celdas o baterías que contienen electrolitos alcalinos (p. ej. las celdas de litio) pueden explotar. Para garantizar la seguridad del producto, las celdas o baterías solo deben ser sustituidas por el tipo Rohde & Schwarz correspondiente (ver lista de recambios).
7. Las baterías y celdas deben reciclarse y no deben tirarse a la basura doméstica. Las baterías o acumuladores que contienen plomo, mercurio o cadmio deben tratarse como residuos especiales. Respete en esta relación las normas nacionales de eliminación y reciclaje.

Transporte

1. El producto puede tener un peso elevado. Por eso es necesario desplazarlo o transportarlo con precaución y, si es necesario, usando un sistema de elevación adecuado (p. ej. una carretilla elevadora), a fin de evitar lesiones en la espalda u otros daños personales.
2. Las asas instaladas en los productos sirven solamente de ayuda para el transporte del producto por personas. Por eso no está permitido utilizar las asas para la sujeción en o sobre medios de transporte como p. ej. grúas, carretillas elevadoras de horquilla, carros etc. Es responsabilidad suya fijar los productos de manera segura a los medios de transporte o elevación. Para evitar daños personales o daños en el producto, siga las instrucciones de seguridad del fabricante del medio de transporte o elevación utilizado.

3. Si se utiliza el producto dentro de un vehículo, recae de manera exclusiva en el conductor la responsabilidad de conducir el vehículo de manera segura y adecuada. El fabricante no asumirá ninguna responsabilidad por accidentes o colisiones. No utilice nunca el producto dentro de un vehículo en movimiento si esto pudiera distraer al conductor. Asegure el producto dentro del vehículo debidamente para evitar, en caso de un accidente, lesiones u otra clase de daños.

Eliminación/protección del medio ambiente

1. Los dispositivos marcados contienen una batería o un acumulador que no se debe desechar con los residuos domésticos sin clasificar, sino que debe ser recogido por separado. La eliminación se debe efectuar exclusivamente a través de un punto de recogida apropiado o del servicio de atención al cliente de Rohde & Schwarz.
2. Los dispositivos eléctricos usados no se deben desechar con los residuos domésticos sin clasificar, sino que deben ser recogidos por separado.
   Rohde & Schwarz GmbH & Co.KG ha elaborado un concepto de eliminación de residuos y asume plenamente los deberes de recogida y eliminación para los fabricantes dentro de la UE. Para desechar el producto de manera respetuosa con el medio ambiente, diríjase a su servicio de atención al cliente de Rohde & Schwarz.
3. Si se trabaja de manera mecánica y/o térmica cualquier producto o componente más allá del funcionamiento previsto, pueden liberarse sustancias peligrosas (polvos con contenido de metales pesados como p. ej. plomo, berilio o níquel). Por eso el producto solo debe ser desmontado por personal especializado con formación adecuada. Un desmontaje inadecuado puede ocasionar daños para la salud. Se deben tener en cuenta las directivas nacionales referentes a la eliminación de residuos.
4. En caso de que durante el trato del producto se formen sustancias peligrosas o combustibles que deban tratarse como residuos especiales (p. ej. refrigerantes o aceites de motor con intervalos de cambio definidos), deben tenerse en cuenta las indicaciones de seguridad del fabricante de dichas sustancias y las normas regionales de eliminación de residuos. Tenga en cuenta también en caso necesario las indicaciones de seguridad especiales contenidas en la documentación del producto. La eliminación incorrecta de sustancias peligrosas o combustibles puede causar daños a la salud o daños al medio ambiente.

Se puede encontrar más información sobre la protección del medio ambiente en la página web de Rohde & Schwarz.

Qualitätszertifikat
Certificate of quality
Certificat de qualité

Certified Quality System
ISO 9001

Certified Environmental System
ISO 14001

Sehr geehrter Kunde,
Sie haben sich für den Kauf eines Rohde & Schwarz-Produktes entschieden. Hiermit erhalten Sie ein nach modernsten Fertigungsmethoden hergestelltes Produkt. Es wurde nach den Regeln unseres Qualitätsmanagementsystems entwickelt, gefertigt und geprüft. Das Rohde & Schwarz-Qualitätsmanagementsystem ist u.a. nach ISO 9001 und ISO 14001 zertifiziert.

Der Umwelt verpflichtet

- Energie-effiziente, RoHS-konforme Produkte
- Kontinuierliche Weiterentwicklung nachhaltiger Umweltkonzepte
- ISO 14001-zertifiziertes Umweltmanagementsystem

Dear Customer,
You have decided to buy a Rohde & Schwarz product. You are thus assured of receiving a product that is manufactured using the most modern methods available. This product was developed, manufactured and tested in compliance with our quality management system standards. The Rohde & Schwarz quality management system is certified according to standards such as ISO 9001 and ISO 14001.

Environmental commitment

- Energy-efficient products
- Continuous improvement in environmental sustainability
- ISO 14001-certified environmental management system

Cher client,
Vous avez choisi d'acheter un produit Rohde & Schwarz. Vous disposez donc d'un produit fabriqué d'après les méthodes les plus avancées. Le développement, la fabrication et les tests respectent nos normes de gestion qualité. Le système de gestion qualité de Rohde & Schwarz a été homologué, entre autres, conformément aux normes ISO 9001 et ISO 14001.

Engagement écologique

- Produits à efficience énergétique
- Amélioration continue de la durabilité environnementale
- Système de gestion de l'environnement certifié selon ISO 14001

ROHDE & SCHWARZ

Customer Support

Technical support – where and when you need it

For quick, expert help with any Rohde & Schwarz equipment, contact one of our Customer Support Centers. A team of highly qualified engineers provides telephone support and will work with you to find a solution to your query on any aspect of the operation, programming or applications of Rohde & Schwarz equipment.

Up-to-date information and upgrades

To keep your instrument up-to-date and to be informed about new application notes related to your instrument, please send an e-mail to the Customer Support Center stating your instrument and your wish. We will take care that you will get the right information.

Europe, Africa, Middle East	Phone +49 89 4129 12345 customersupport@rohde-schwarz.com
North America	Phone 1-888-TEST-RSA (1-888-837-8772) customer.support@rsa.rohde-schwarz.com
Latin America	Phone +1-410-910-7988 customersupport.la@rohde-schwarz.com
Asia/Pacific	Phone +65 65 13 04 88 customersupport.asia@rohde-schwarz.com
China	Phone +86-800-810-8228 / +86-400-650-5896 customersupport.china@rohde-schwarz.com

ROHDE&SCHWARZ

Contents

1 Preface

1.1 Documentation Overview

The user documentation for the R&S WinIQSIM2 consists of the following parts:

- Online Help system on the instrument,
- Documentation CD-ROM with:
  - Online help system (*.chm) as a standalone help,
  - Software Manual and Operating Manuals for the options,
  - Data sheet and specifications,
  - Links to useful sites on the R&S internet.

Online Help

The Online Help is embedded in the instrument's firmware. It offers quick, context-sensitive access to the complete information needed for operation and programming. The online help contains help on operating the R&S WinIQSIM2 and all available options.

Software Manual and Operating Manuals

These manuals are available in PDF format - in printable form - on the Documentation CD-ROM delivered with the software. In the Software Manual, all instrument functions are described in detail. Furthermore, it provides an introduction to remote control and a complete description of the remote control commands with programming examples. Information on instalation, interfaces and error messages is also given.

In the individual option manuals, the specific instrument functions of the option are described in detail. For additional information on default settings and parameters, refer to the data sheets. Basic information on operating the R&S WinIQSIM2 is not included in the option manuals.

The following Operating Manuals are available for the R&S WinIQSIM2:

- Digital Standard GSM/EDGE, incl. EDGE+
- Digital Standard 3GPP FDD, 3GPP FDD enhanced MS/BS tests incl. HSDPA, 3GPP FDD HSUPA, 3GPP FDD HSPA+
- Digital Standard GPS and Assisted GPS
- Digital Standard CDMA2000® incl. 1xEV-DV
- Digital Standard 1xEV-DO Rev. A
- Digital Standard IEEE 802.11 (a/b/c) WLAN
- Digital Standard IEEE 802.16 WiMAX
- Digital Standard TD-SCDMA, inclusive TD-SCDMA enhanced features
- Digital Standard DVB-H
- Digital Standard DAB / T-DMB
- Digital Standard IEEE 802.11n WLAN-N

- Digital Standard EUTRA/LTE
- Digital Standard TETRA Release 2
- Digital Standard Bluetooth®Enhanced Data Rate

These manuals can also be orderd in printed form (see ordering information in the data sheet).

Release Notes

The release notes describe new and modified functions, eliminated problems, and last minute changes to the documentation. The corresponding firmware version is indicated on the title page of the release notes. The current release notes are provided in the Internet.

1.2 Typographical Conventions

The following text markers are used throughout this documentation:

Convention	Description
"Graphical user interface elements"	All names of graphical user interface elements on the screen, such as dialog boxes, menus, options, buttons, and softkeys are enclosed by quotation marks.
KEYS	Key names are written in capital letters.
`File names, commands, program code`	File names, commands, coding samples and screen output are distinguished by their font.
Input	Input to be entered by the user is displayed in italics.
Links	Links that you can click are displayed in blue font.
"References"	References to other parts of the documentation are enclosed by quotation marks.

2 Signal Generation Software R&S WinIQSIM2

R&S WinIQSIM2 permits calculation of I and Q baseband signals on a Windows PC system. It combines the advantages and functionalities of the well known previous version of R&S WinIQSIM with the firmware of the R&S SMU/SMATE vector signal generator family.

Among R&S WinIQSIM2' functions are single-carrier modulation, the generation of multi carrier, W-CDMA and other 3rd Generation signals. Future digital standards will be supported as well.

The look-and-feel of R&S WinIQSIM2, especially the block diagram, is based on the R&S SMU firmware. So users of R&S SMU vector signal generators can easily get used to the handling and behavior of this software. Other users will find an easy approach as well, since the software offers a high grade of self-explanation by means of a logically structured signal flow diagram and menus.

The whole process of signal configuration can be controlled via the block diagram. The current status of signal configuration and active interfering components can be seen at a glance.

Comprehensive graphic display modes in the time and frequency domain, e.g., I(t) and q(t), vector diagram or spectrum, allow simulation and analysis of characteristics as early as in the design phase of new digital communications systems.

All in all, R&S WinIQSIM2 is an indispensable application for anyone engaged in state-of-the-art digital modulation.

Generation of waveforms for ARB generators

Waveform signals computed by the R&S WinIQSIM2 can be loaded onto and processed by the following ARB generators:

- R&S AMU
- R&S AFQ
- R&S CMW
- R&S SMU
- R&S SMJ
- R&S SMATE
- R&S SMBV
- R&S EX-IQ-BOX
  (waveform generation without data transfer)

Control of vector signal generators

The following vector signal generators can be used for generation of radio frequency (RF) signals. Moreover, these vector signal generators can be controlled by the R&S WinIQSIM2:

- R&S SMU
- R&S SMJ

- R&S SMATE
- R&S SMBV

3 Installation of R&S WinIQSIM2 Software

R&S WinIQSIM2 is delivered in the latest version available on the R&S WinIQIM2 CD-ROM. Software updates as well as the Release Notes describing the improvements and modifications are provided on the Internet at the download site of the Rohde&Schwarz home page. This home page always offers the latest information on your R&S WinIQSIM2 as well as the signal generator software, e.g., also on changes of the software update procedure.

To uninstall and to install R&S WinIQSIM2, the following hardware and software requirements have to be met.

Software Requirements

One of the following operating systems: ● Windows XP SP2 ● Windows Vista ● Window 7	R&S WinIQSIM2 has to be installed on one of the supported operating systems. **Note:** Any other Windows version or other operating systems are not supported. During installation, the operation system is checked. The installation will be terminated if this requirement is not fulfilled.
National Instruments (NI) VISA version 3.4.1 or higher (optional) or Agilent VISA version 14 or higher (optional)	VISA driver is usually bundled with a GPIB (IEC/IEEE)-bus card from National Instruments (NI) or Agilent. If none of these cards is used, there is the option to perform data transmission via TCP/IP connection over LAN. Drivers can be obtained from R&S or NI directly.

Hardware Requirements

CPU	At least Pentium or compatible, as from 1 GHz (recommended).
Virtual Memory	At least 1GB. During installation the virtual memory is checked. The installation will be terminated if this requirement is not fulfilled.
VGA color display resolution	At least 800*600 pixels
GPIB (IEC/IEEE)-bus card (optional)	From National Instruments or Agilent.

R&S WinIQSIM2 Software

The R&S WinIQSIM2 software consists of the file `WinIQSIM2_<V.VV.VVV.VV>.exe`. The version numbers in the file names vary with each update.

3.1 Installing R&S WinIQSIM2

Uninstall an old software version

Skip the following, if this is a first-time installation.

► To uninstall a previous version, go to "Start > Settings > Control Panel > Add/Remove Programs" and select the entry `WinIQSIM2_V.VV.VVV.VV`.

The script file identifies and removes all currently installed R&S signal generator software items.

Install a new software version

ⓘ Administrator rights are necessary for installation and starting.

► In Windows Explorer double-click `WinIQSIM2_V.VV.VVV.VV.exe` and follow the instructions.

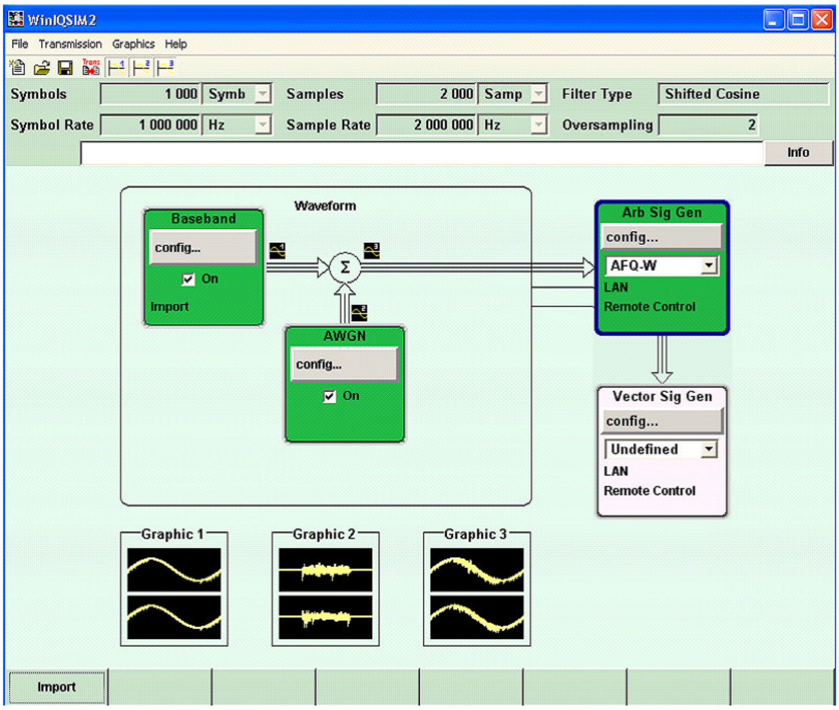
4 Getting Started

An overview of the operating menu and of the toolbar in the Main window of R&S WinIQSIM2 is given in chapter 5, "Graphical User Interface", on page 25.

4.1 System Block Diagram

This section gives an overview of control elements, connectors, and signal configuration in the Main window. It shows the block diagrams, i.e., the whole system including connected devices at a glance. The graphical elements can be accessed for operation. Required menus and graphs are displayed in the block diagram, which contains the internal signal flow from baseband on the left position to the Signal Generator Blocks at the right. Each element/connector is briefly described.

The signal flow between the Signal Generator Blocks and the employed inputs and outputs are also represented.



Symbols 1 000 Symb
Samples 2 000 Samp
Filter Type Shifted Cosine
Symbol Rate 1 000 000 Hz
Sample Rate 2 000 000 Hz
Oversampling 2

The Winbar at the bottom of the block diagram gives an overview of open dialogs and simplifies their access.

Labelled buttons represent open dialogs, the label indicates the dialogs. If several dialogs are open, the button of the currently active dialog is displayed in a lighter color. Up to eight dialogs may be open simultaneously. When the ninth dialog is opened, the dialog that was opened first is automatically closed.

For more information see

4.2 Control and Operation of R&S WinIQSIM2 via Block Diagram

Each of the blocks in the Main window contributes to the signal flow. Here, R&S WinIQSIM2 is controlled, the operational tasks can be performed, and the signal flow is visualized.

Selecting a digital standard

1. In the "Baseband" block diagram, below the config button, the state check box is located. Here, the states On or Off of the component block are controlled. By activating the block, its color switches to green and indicates that the block contributes to the signal flow.
2. Click the "config" button.
   A menu with a list of optional standards opens.

3. By selecting one of the offered options, a dialog opens. Parameter entries for the selected digital standard can be made.

4.3 Signal Configuration Examples

This section describes the configuration procedure for a single-carrier signal with AWGN by way of an example.

To generate an I/Q signal with AWGN, the following steps are required:

- chapter 4.3.1, "Configuring an Instrument", on page 16
- chapter 4.3.2, "Selecting an Instrument", on page 18
- chapter 4.3.3, "Generating the Waveform", on page 18
- chapter 4.3.4, "Displaying and Configuring the Graphical Representation of the Waveform Signal", on page 19
- chapter 4.3.5, "Adding Noise (AWGN) to the Signal and Displaying the Signal Flow", on page 20
- chapter 4.3.6, "Transmitting the Signal to the R&S AFQ", on page 22

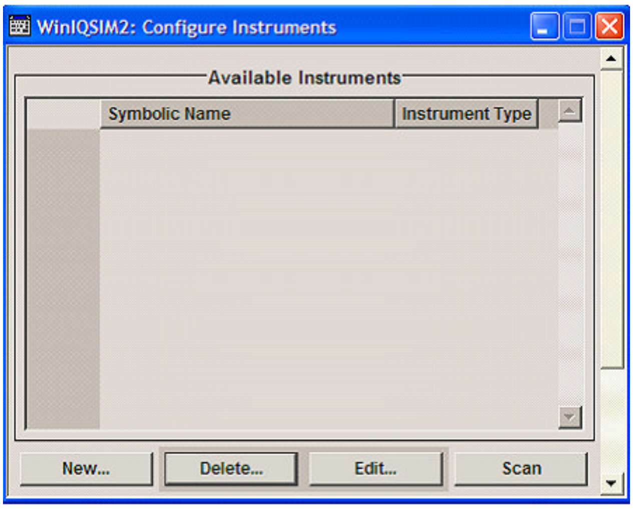
4.3.1 Configuring an Instrument

There are two options:

- No instrument has been created yet.
  It is necessary to create and select an instrument before signal generation is possible. A dummy instrument might be created.
- An instrument has already been created and selected.
  Continue in chapter 4.3.3, "Generating the Waveform", on page 18.

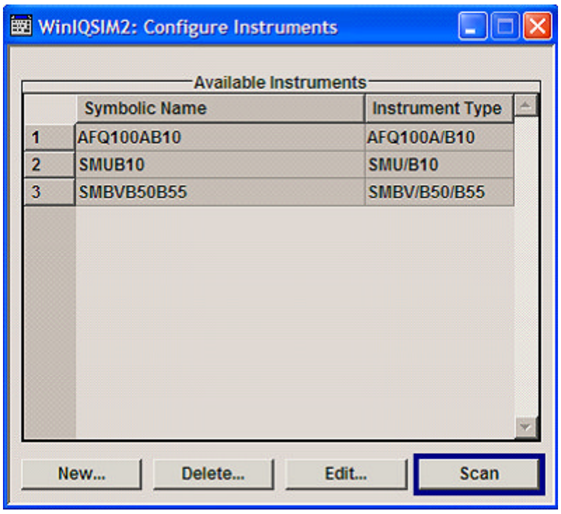
1. Connect the instrument (in this example an R&S AFQ100A with option B10) via LAN interface and switch it on.
2. Open the Configure Instruments dialog by selecting "Transmission > Instruments".

   The "Configure Instruments" dialog opens.



   In this dialog, you create the instrument or create a dummy instrument and configure the connection to the selected instrument.
   R&S WinIQSIM2 allows you to scan the LAN, GPIB, or USB interfaces for connected instruments. Instruments found during the scan are listed in the list "Available Instruments" and can be selected for configuration.
3. Click the "Scan" button to scan the network for instruments.

   All instruments found in the network as well as instruments connected via the GPIB or USB interfaces are displayed in the "Available Instruments" list. R&S WinIQSIM2 obtains all information for connecting to the instrument, so further configuration is not necessary but can be performed via the "Edit" button, if desired.



	Symbolic Name	Instrument Type
1	AFQ100AB10	AFQ100A/B10
2	SMUB10	SMU/B10
3	SMBVB50B55	SMBV/B50/B55

Tip: R&S WinIQSIM2 does not clear the list "Available Instruments" before a scan. Any instruments found during a scan will be added to the list.

If the instrument of your choice is in the "Available Instruments" list, close the dialog. Now the R&S AFQ is available for selection in the "Arb Sig Gen" block.

If the instrument of your choice is not displayed in the "Available Instruments" list, check if the following prerequisites are fulfilled, and scan for the instrument again:

- Is the instrument switched on?
- Is the instrument accessible via LAN, GPIB, or USB?
- Is the firewall of the instrument's operating system deactivated?

For more information, see the following sections:

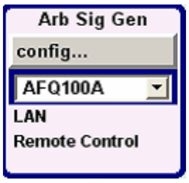
- chapter 4.3.2, "Selecting an Instrument", on page 18
- chapter 4.3.6, "Transmitting the Signal to the R&S AFQ", on page 22

4.3.2 Selecting an Instrument

Waveform calculation depends on the instrument settings, so before you can create the waveform, the instruments has to be selected first. For signal calculation, it is not necessary that an instrument is connected. The signal calculation can be performed for a dummy instrument (offline), but even this dummy instrument has to be selected before waveform calculation.

The instrument has to be configured before it can be selected. For configuring the instrument, see chapter 4.3.1, "Configuring an Instrument", on page 16.

► In the "Main" window, in the "Arb Sig Gen" block, select the R&S AFQ from the combo box.

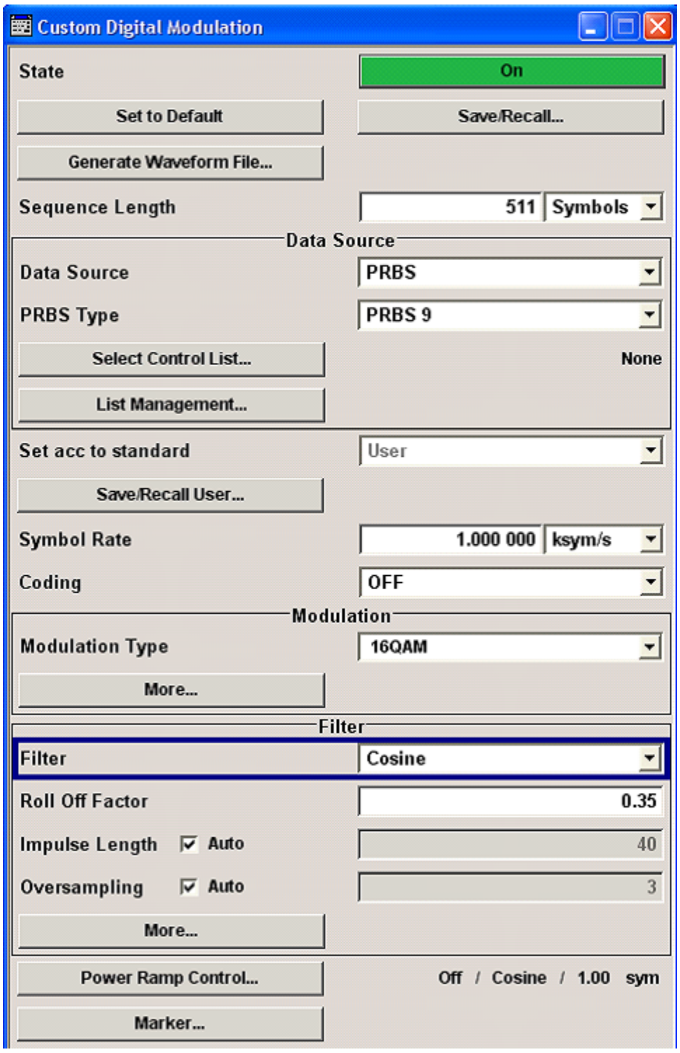


The R&S AFQ100A is selected. The prerequisites for waveform calculation are fulfilled.

4.3.3 Generating the Waveform

The example task is to create a 16-QAM encoded waveform. The data consists of a sequence length of 511 samples of a pseudo random binary sequence (shift register length: 9 Bits). The symbol rate is set at 1 ksym/s. A cosine filter is used.

1. In the baseband block at the left position, click "Config > Custom Digital Mod".



State: On

Sequence Length: 511 Symbols

Data Source: PRBS

PRBS Type: PRBS 9

Select Control List...: None

Set acc to standard: User

Symbol Rate: 1.000 000 ksym/s

Coding: OFF

Modulation Type: 16QAM

Filter: Cosine

Roll Off Factor: 0.35

Impulse Length ☑ Auto: 40

Oversampling ☑ Auto: 3

Power Ramp Control...: Off / Cosine / 1.00 sym

2. Make the following entries:
   - "Sequence Length" = 511 Symbols
   - "Symbol Rate" = 1.000 000 ksym/s
   - "Modulation Type" = 16QAM
   - "Filter" = Cosine

3. Click the "State" button to change the state to On.

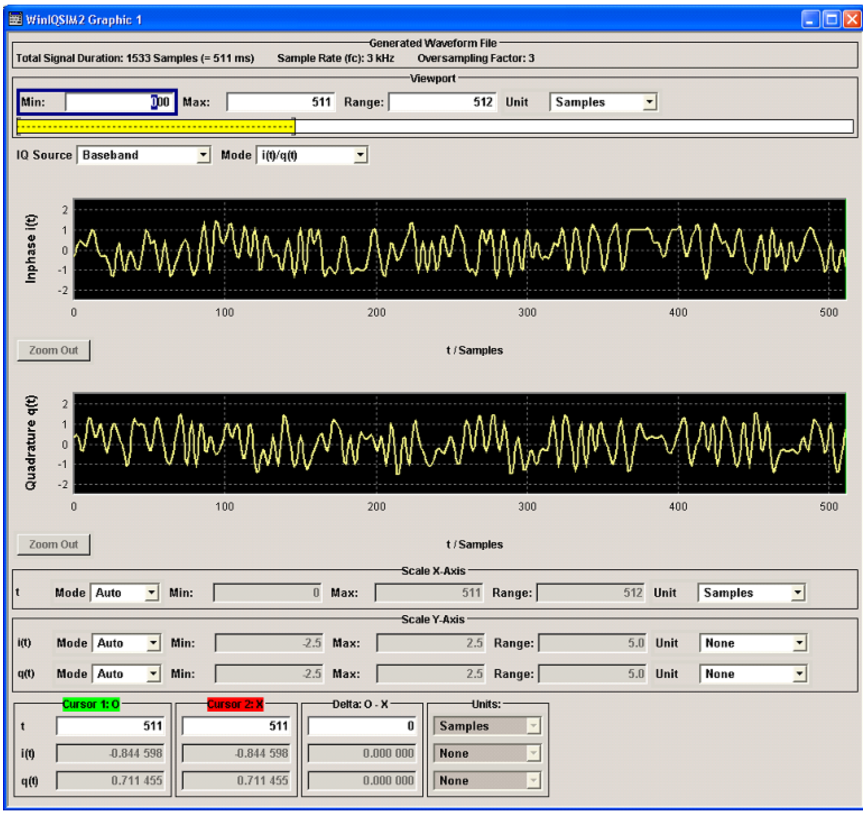
   A bar appears indicating the calculation progress. The waveform is generated. It can be checked by means of a graphic tool.

4.3.4 Displaying and Configuring the Graphical Representation of the Waveform Signal

1. In the Main window, select "Graphics > Graphic 1 > Graphic 1 (Preview Only)".

A small graphics display labelled "Graphic 1" opens at the bottom of the block diagram. A small icon shows up in the signal flow path indicating the position from where the graphics obtains its data.

2. Double-click the small graphics display to open the graphics dialog.



Generated Waveform File

Total Signal Duration: 1533 Samples (= 511 ms) Sample Rate (fc): 3 kHz Oversampling Factor: 3

Viewport

Min: 000 Max: 511 Range: 512 Unit Samples

IQ Source Baseband Mode i(t)/q(t)

Scale X-Axis

t Mode Auto Min: 0 Max: 511 Range: 512 Unit Samples

Scale Y-Axis

i(t) Mode Auto Min: -2.5 Max: 2.5 Range: 5.0 Unit None

q(t) Mode Auto Min: -2.5 Max: 2.5 Range: 5.0 Unit None

	Cursor 1: O	Cursor 2: X	Delta: O - X	Units:
t	511	511	0	Samples
i(t)	-0.844 598	-0.844 598	0.000 000	None
q(t)	0.711 455	0.711 455	0.000 000	None

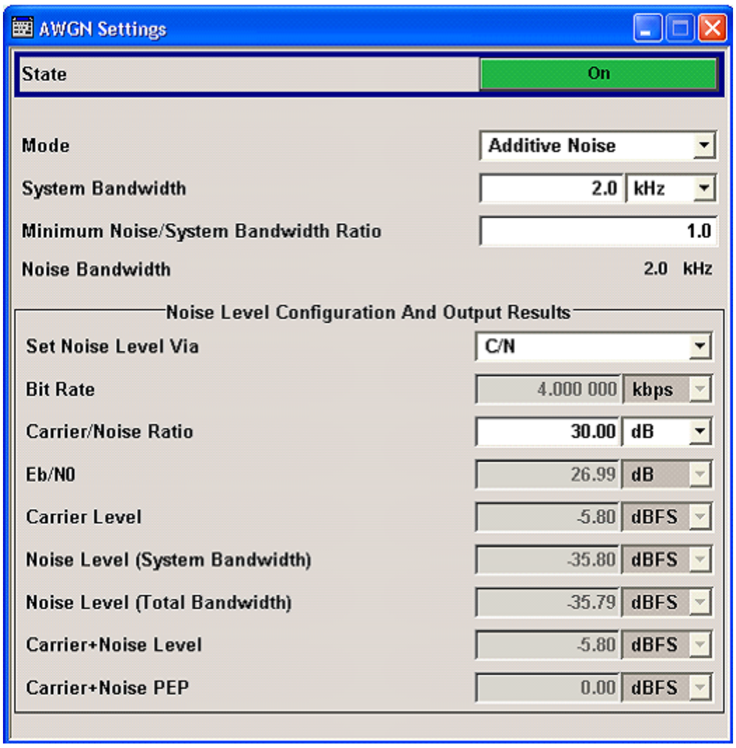
In the "Mode" combo box, the graphical representation mode can be selected, e.g., I/Q waveform format, vector diagram, spectrum, and many more.
The graphics panel is an important tool for viewing and checking the signal configuration . In this panel, the generated I/Q signal or representations derived thereof like the vector diagram or the signal spectrum can be displayed.

For more information, see chapter 5.7, "Graphics Menu", on page 35

4.3.5 Adding Noise (AWGN) to the Signal and Displaying the Signal Flow

1. Select "Main window > AWGN > Config > AWGN".

   The "AWGN Settings" dialog opens.



State	On	
Mode	Additive Noise	
System Bandwidth	2.0	kHz
Minimum Noise/System Bandwidth Ratio	1.0	
Noise Bandwidth	2.0	kHz

Noise Level Configuration And Output Results

Set Noise Level Via	C/N	
Bit Rate	4.000 000	kbps
Carrier/Noise Ratio	30.00	dB
Eb/N0	26.99	dB
Carrier Level	-5.80	dBFS
Noise Level (System Bandwidth)	-35.80	dBFS
Noise Level (Total Bandwidth)	-35.79	dBFS
Carrier+Noise Level	-5.80	dBFS
Carrier+Noise PEP	0.00	dBFS

2. Make the following entries:
   - "System Bandwidth" = 2 kHz
   - "Carrier/Noise Ratio" = 30 dB
3. Click the "State" button to change the state to On.

   A bar appears indicating the calculation progress. The AWGN is generated. It can be checked by means of a graphic tool.
4. To display the AWGN signal, in the Main window, select "Graphics", point to "Graphic 2", and click "Graphic 2 (Preview Only)".
   A small graphics display labelled "Graphic 2" opens at the bottom of the block diagram. A small icon shows up in the signal flow path indicating the position from where the graphics obtain information.
5. Double-click the small graphics display to open the graphics dialog.
6. Make the following entries:
   - "IQ Source" = AWGN
   - "Mode" = FFT Magnitude
7. To display the graphics dialog for the baseband signal, in the Main window, select "Graphics", point to "Graphic 1", and click "Graphic 1 (Complete)".

   The "Graphics" dialog for the baseband signal opens.
8. Make the following entries:
   - "IQ Source" = Baseband
   - "Mode" = FFT Magnitude

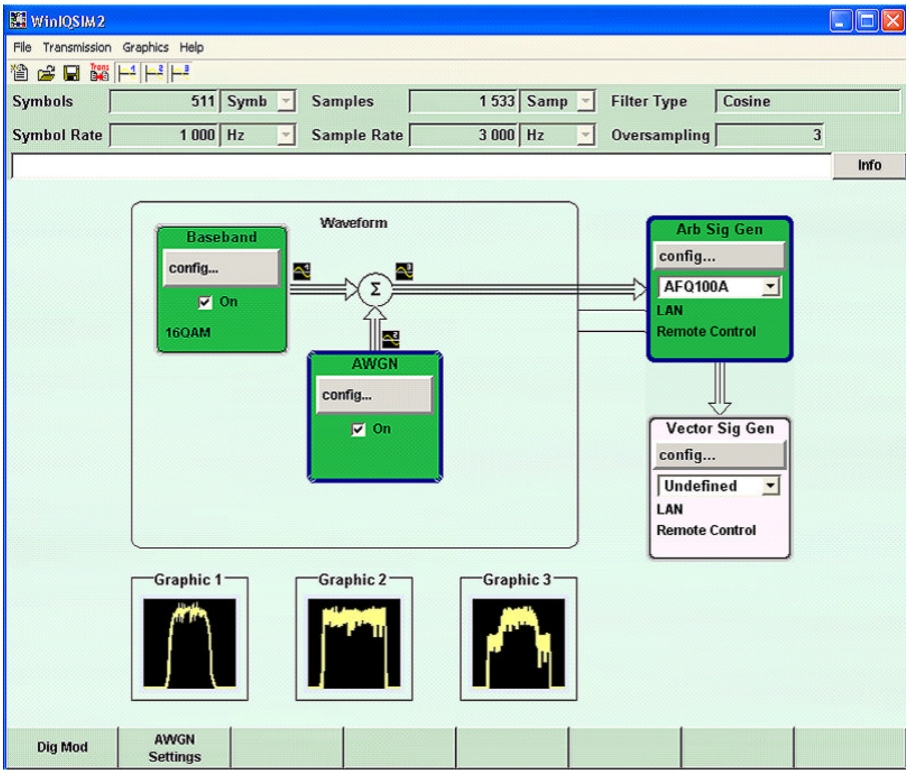
9. To display the baseband and AWGN signals, in the Main window, select "Graphics", point to "Graphic 3", and click "Graphic 3 (Complete)".

   The "Graphics" dialog opens.

10. Make the following entries:
    - "IQ Source" = Baseband+AWGN
    - "Mode" = FFT Magnitude

11. Close all graphics dialogs.

    You can now see the graphical representation before summation with AWGN, with AWGN only, and after the summation point (noise afflicted baseband signal).

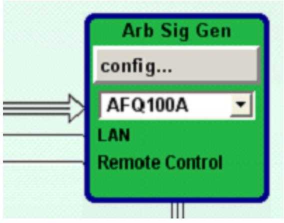


Symbols 511 Symb
Samples 1 533 Samp
Filter Type Cosine
Symbol Rate 1 000 Hz
Sample Rate 3 000 Hz
Oversampling 3

Baseband: ☑ On, 16QAM
AWGN: ☑ On
Arb Sig Gen: AFQ100A, LAN, Remote Control
Vector Sig Gen: Undefined, LAN, Remote Control

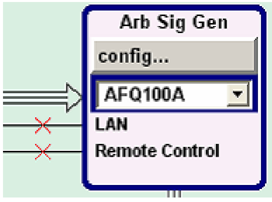
4.3.6 Transmitting the Signal to the R&S AFQ

Before you can transmit the waveform to the R&S AFQ, you have to ensure that the connection to the R&S AFQ is enabled. This is indicated by two thin lines leading from the R&S WinIQSIM2 block to the "Arb Sig Gen" block.



Arb Sig Gen: AFQ100A, LAN, Remote Control

In case of crossed lines, the connection has to be established before the waveform can be transmitted to the instrument.



The prerequisites for a successful connection to the instrument are:

- Instrument is switched on
- Instrument accessible either via LAN, GPIB, or USB
- LAN connection is enabled in the instrument's operating system
- Firewall is off in the instrument's operating system

By default, the firewall is on and LAN connection is disabled, so you have to change the settings.

There are two ways to change the settings:

- When starting up the R&S AFQ
- Via Control Panel of the instrument's operating system

To change the settings when starting up the R&S AFQ:

1. Switch off the R&S AFQ, and switch it on again.
2. When during startup the status LEDs on the R&S AFQ's front panel start to flash counterclockwise, push the TRIGGER button of the R&S AFQ.
   This disables the firewall and enables Remote Desktop of the R&S AFQ's operating system.
   When you connect to the instrument via Remote Desktop, you are asked if you want to either permanently disable the firewall and enable the Remote Desktop or for the current session only.

   Now the R&S AFQ is ready to play and the waveform can be created.

To disable the firewall of the R&S AFQ via Control Panel:

1. Select "Start > Control Panel > Windows Firewall".

   The Windows Firewall dialog opens.
2. On the "General" tab, activate Off (not recommended).
3. Click OK.

   The Windows Firewall is disabled.

To enable Remote Desktop of the R&S AFQ via Control Panel:

1. Select "Start > Control Panel > System".

   The "System Properties" dialog opens.

2. Click the "Remote" tab, and activate "Allow" users to connect remotely to this computer.

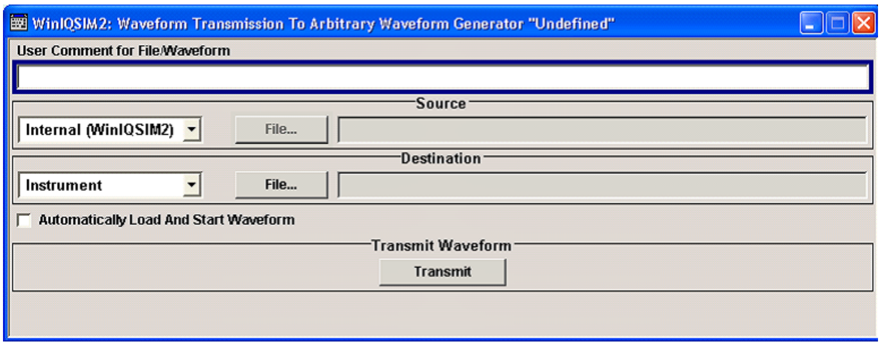
3. Click OK.

   Remote Desktop is enabled.

To transmit the signal to the R&S AFQ, perform as follows:

1. In the Main window, in the tool bar, click the "Transmit" button .

   The "Waveform Transmission To Arbitrary Signal Generator" dialog opens.



WinIQSIM2: Waveform Transmission To Arbitrary Waveform Generator "Undefined"

User Comment for File/Waveform

Source: Internal (WinIQSIM2)

Destination: Instrument

□ Automatically Load And Start Waveform

2. In the "Source" group box, select "Internal (WinIQSIM2)". The file generated latest is chosen.

3. In the "Destination" group box, choose "Instrument" and click the "File" button.

   A file browser dialog of the destination device opens to prompt the destination file name.
   Type in a file name and click OK.

4. Enable the "Automatically Load and Start Waveform" check box.

   After the waveform has been successfully transmitted to the device, the signal will immediately be generated by the R&S AFQ.

5. Click the "Transmit" button to start the transmission.

   A progress bar indicates the status of the transmission process. If the transmission has been successful, the waveform will be loaded into the ARB and is available at the output connectors of the R&S AFQ.

5 Graphical User Interface

In the following, a detailed description of the R&S WinIQSIM2 user interface is given. The Main window with the block diagram is the operating and control interface for the whole program. From here, all program functions are accessible. This panel is displayed after the start of R&S WinIQSIM2. The program always loads the previously used settings so you can continue your work in the next session.

5.1 Operating Menu and Toolbar in the Main Window

At the top of the Main window of R&S WinIQSIM2, there are the menu bar, the tool bar, and the Info bar with its corresponding Info button.

To offer the user a familiar environment, operation is very similar to operation of MS Windows user interfaces. All menus and dialogs are built using known elements, e.g., selection lists, check boxes, and entry fields.

A blue frame indicates that the selected item is active. In the highlighted element, entries can be made.

Table 5-1: Content of the menu

File	**Transmission**	**Graphics**	**Help**
New	Instruments	Graphic 1	Contents
Open	Transmit	Graphic 2	Index
Save		Graphic 3	
Save as			
Setup			
Exit			

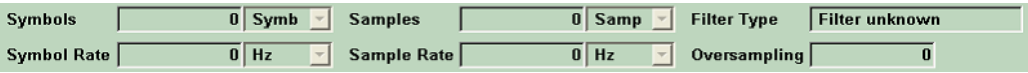
The main menu functions are also accessible via the toolbar with its icons below the menu selection line.

Table 5-2: Toolbar icons in R&S WinIQSIM2

📄	New	Resets the system to default.
📂	Open	Loads a previously stored configuration file containing all settings.
💾	Save as	Stores all settings made by R&S WinIQSIM2 in a configuration file.
Trans	Transmit	Opens the Transmit dialog.
1	Graphics	Activates the small graphics display at the bottom of the diagram.

5.2 Signal Information Bar in the Main Window

Below the toolbar, the Signal Information bar is located.



Symbols 0 Symb Samples 0 Samp Filter Type Filter unknown
Symbol Rate 0 Hz Sample Rate 0 Hz Oversampling 0

In the "Signal Information" bar, the following fields and boxes are displayed:

"Symbols"	Number of symbols.
"Symbol Rate"	Rate (symbols/sec) at which the symbols are sent.
"Samples"	Number of samples, respectively the number of symbols times over sampling factor.
"Sample Rate"	Rate (samples/sec) at which the samples are sent, respectively the symbol rate times over sampling factor.
"Filter Type"	Type of the filter that is applied to the signal.
"Oversampling"	Factor by which the symbols have been upsampled to obtain the waveform samples.

5.3 Messages in the Info Bar

Messages indicate information, warnings, and errors. They are displayed in the info line in different colors depending on their importance and display duration. A few operating states and the current messages are displayed in the info line. For information on messages in greater detail and their management, an Info dialog can be opened.

The following messages are displayed:

- Error
  An error, e.g., settings conflict, is displayed in red color.
  There are two options:
  - Critical:
    Prevents R&S WinIQSIM2 from working, e.g., a DLL file version does not match. Critical errors are displayed in red color.
  - System:
    Concerns the operating system, e.g., wrong file path. System errors are displayed in black color.
- Information
  The information, e.g., file not found, is displayed in black color.
- Warning
  A warning indicates a less significant error and is displayed in black color.
- Brief message
  Brief messages report automatic settings in the program, e.g., switching off of incompatible types of modulation, or on illegal entries that are not accepted by the program, e.g., range violations. They are displayed in the info line on a yellow background. They are displayed on top of status information or permanent messages.

Brief messages usually do not demand user actions and disappear automatically after a short period of time. They are stored in the history, however.

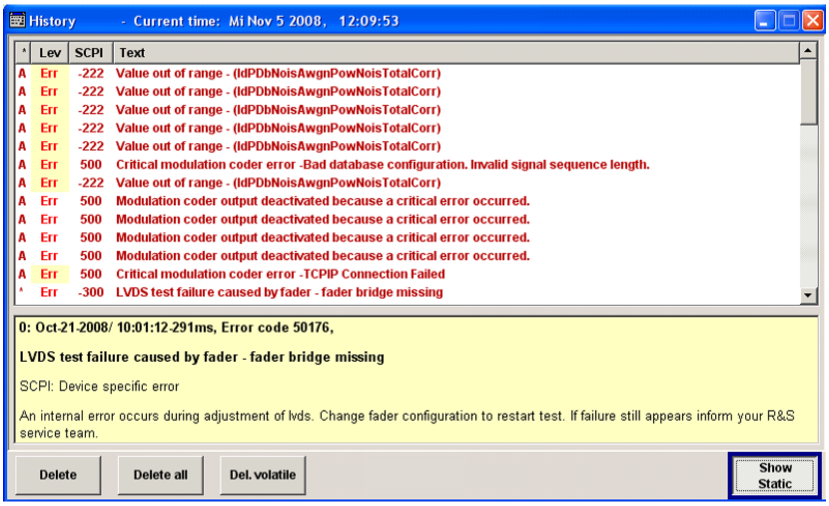
- Permanent messages
  Permanent messages are displayed if an error occurs that impairs further program operation. The error signaled by a permanent message must be eliminated before correct software operation can be ensured.
  The message is displayed until the error is eliminated. It covers the status display in the info line. After error elimination, the message automatically disappears and is also recorded in the history.

5.4 Managing Messages in the Info Dialog

A few operating states and the current messages are displayed in the info line. For information on messages in greater detail and their management, an info dialog can be opened.

► In the Main window, click the "Info" button.
  A list of currently active permanent messages and a detailed description of each message is shown. The messages are color-coded according to their level.

  The Info dialog opens.



History - Current time: Mi Nov 5 2008, 12:09:53

*	Lev	SCPI	Text
A	Err	-222	Value out of range - (IdPDbNoisAwgnPowNoisTotalCorr)
A	Err	-222	Value out of range - (IdPDbNoisAwgnPowNoisTotalCorr)
A	Err	-222	Value out of range - (IdPDbNoisAwgnPowNoisTotalCorr)
A	Err	-222	Value out of range - (IdPDbNoisAwgnPowNoisTotalCorr)
A	Err	-222	Value out of range - (IdPDbNoisAwgnPowNoisTotalCorr)
A	Err	500	Critical modulation coder error -Bad database configuration. Invalid signal sequence length.
A	Err	-222	Value out of range - (IdPDbNoisAwgnPowNoisTotalCorr)
A	Err	500	Modulation coder output deactivated because a critical error occurred.
A	Err	500	Modulation coder output deactivated because a critical error occurred.
A	Err	500	Modulation coder output deactivated because a critical error occurred.
A	Err	500	Modulation coder output deactivated because a critical error occurred.
A	Err	500	Critical modulation coder error -TCPIP Connection Failed
*	Err	-300	LVDS test failure caused by fader - fader bridge missing

0: Oct-21-2008/ 10:01:12-291ms, Error code 50176,

LVDS test failure caused by fader - fader bridge missing

SCPI: Device specific error

An internal error occurs during adjustment of lvds. Change fader configuration to restart test. If failure still appears inform your R&S service team.

In the "Info" dialog, in the upper section, the following columns are displayed:

"*"	Signal path The following signal path might occur: ● A: Path A ● * : No path is assigned
"LEV"	Message level. Messages referring to a logical component of WinIQSIM2, e.g., Unicode, are marked in red color, info messages are marked in black color. The following levels might occur: ● Err: Error message ● Info: Information message ● Sys: System message ● Crit: Critical message For more information see chapter 5.3, "Messages in the Info Bar", on page 26.
"SCPI"	Indicates the SCPI error code.
Text	A list of all currently permanent messages in the order of their occurrence, i.e., the most recent message is displayed first.

With the following buttons, error messages can be cleared and a history of all messages can be displayed:

"Delete"	Clears the highlighted message. This button is available only if the history of the messages is displayed.
"Delete All"	Clears all messages. This button is available only if the history of the messages is displayed.
"Del. volatile"	Clears all brief messages. This button is available only if the history of the messages is displayed.
"History"	Calls the list of all messages that have occurred since instrument switch-on. The most recent messages are displayed at the top of the list. When the button is pressed again, the list of current messages is displayed.

Getting Additional Information on the Message

► In the "Info" dialog, click a message to select it.

In the lower section of the dialog, additional information on the highlighted message is displayed.

Displaying all Messages

► In the "Info" dialog, click the "History" button.

A history of all messages that have occurred since R&S WinIQSIM2 software was started is listed in the upper dialog pane. The most recent message is displayed first.

Deleting an Error Message

► In the "Info" dialog, select the highlighted message and click the "Delete" button.

Tip: This button is available only when the history of the messages is displayed.

The highlighted message is cleared.

Deleting all Error Messages

► In the "Info" dialog, click the "Delete All" button.

Tip: This button is available only when the history of the messages is displayed.

All messages are cleared.

Deleting all Brief Messages

► In the "Info" dialog, click the "Del. volatile " button.

Tip: This button is available only when the history of the messages is displayed.

All brief messages are cleared.

Calling the History

1. In the "Info" dialog, click the "History" button.

   A list of all messages that have occurred since instrument switch-on is displayed. The most recent messages are displayed at the top of the list.

2. Click the "History" button once more.
   A list of current messages is displayed.

5.5 File Menu

In the following, the File menu of the R&S WinIQSIM2 Main window is described in detail. It incorporates the standard submenus and additionally a "Setup" submenu.

5.5.1 Standard Functions

New
Resets R&S WinIQSIM2 to default.

Open
Opens the standard file open browser for loading a saved R&S WinIQSIM2 file (`*.savrcl`). The file contains the user specific settings of a session, such as baseband settings, instrument settings, graphic display settings, etc. The complete settings of a session can be saved and loaded.

Only files of this type are selectable.

Save
Standard quick save of the settings of the current session if a file name previously has been applied. If not, the Save as dialog is opened.

Save as
Opens the standard file save browser for saving the settings of the current session. R&S WinIQSIM2 files have the file extension .savrcl so the name typed in is equipped with this extension. The complete settings of a session are saved.

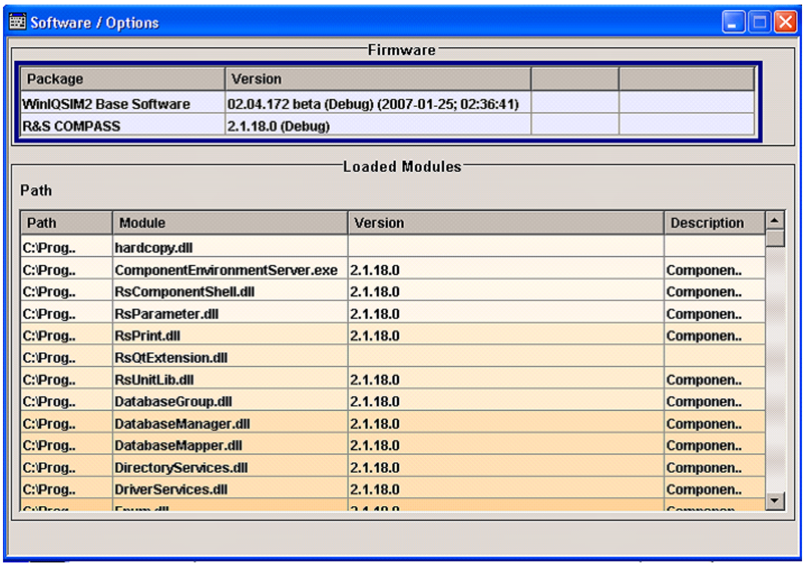
Exit
Quits R&S WinIQSIM2. The current settings of the WinIQSIM2 session are saved and loaded by default when starting WinIQSIM2 again.

5.5.2 Setup

5.5.2.1 Software/Options

► From the "File" menu, select "Setup > Software/Options".

An info dialog opens, showing program information.



Firmware

Package	Version		
WinIQSIM2 Base Software	02.04.172 beta (Debug) (2007-01-25; 02:36:41)		
R&S COMPASS	2.1.18.0 (Debug)		

Loaded Modules

Path

Path	Module	Version	Description
C:\Prog..	hardcopy.dll		
C:\Prog..	ComponentEnvironmentServer.exe	2.1.18.0	Componen..
C:\Prog..	RsComponentShell.dll	2.1.18.0	Componen..
C:\Prog..	RsParameter.dll	2.1.18.0	Componen..
C:\Prog..	RsPrint.dll	2.1.18.0	Componen..
C:\Prog..	RsQtExtension.dll		
C:\Prog..	RsUnitLib.dll	2.1.18.0	Componen..
C:\Prog..	DatabaseGroup.dll	2.1.18.0	Componen..
C:\Prog..	DatabaseManager.dll	2.1.18.0	Componen..
C:\Prog..	DatabaseMapper.dll	2.1.18.0	Componen..
C:\Prog..	DirectoryServices.dll	2.1.18.0	Componen..
C:\Prog..	DriverServices.dll	2.1.18.0	Componen..

Firmware
In the group box "Firmware", the following columns are displayed:

"Package"	Installed software packages.
"Version"	Release of the software package.

Loaded Modules
In the group box "Loaded Modules", the following columns are displayed:

"Path"	Path to the file location.

"Module" A module file.

"Version" Version of a file.

"Description" Description of the module file.

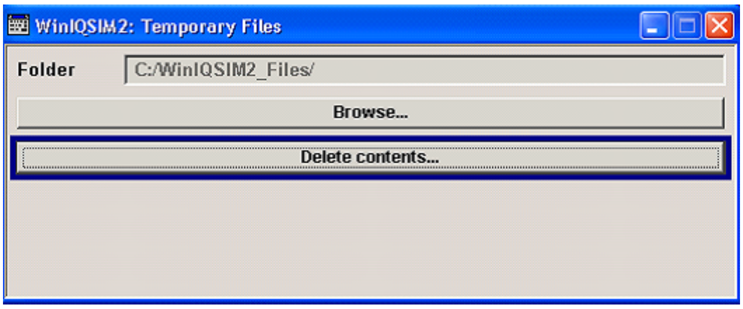
Note: In the "Description" column, double-click a cell to get detailed information. For your convenience, the rows have a colored background.

5.5.2.2 Temporary Files

You can define a location where you want to save the R&S WinIQSIM2 temporary files.

1. From the "File" menu, select "Setup > Temporary Files".

   The "Temporary Files" dialog opens.



Folder C:/WinIQSIM2_Files/

2. In the "Folder" field, the currently selected location for temporary R&S WinIQSIM2 files is displayed.
3. To change the folder, click "Browse".

   The "Browse For Folder" dialog opens.
4. Select the folder of your choice, and click OK.

   The selected folder is displayed with the complete path in the "Folder" field.
5. To delete the contents in the folder, click "Delete Contents".

   A message box opens.
6. Click OK to delete all the files in the folder.

5.6 Transmission Menu

In the following, all settings that are responsible for transmission of the waveform file from R&S WinIQSIM2 to the instrument are described.

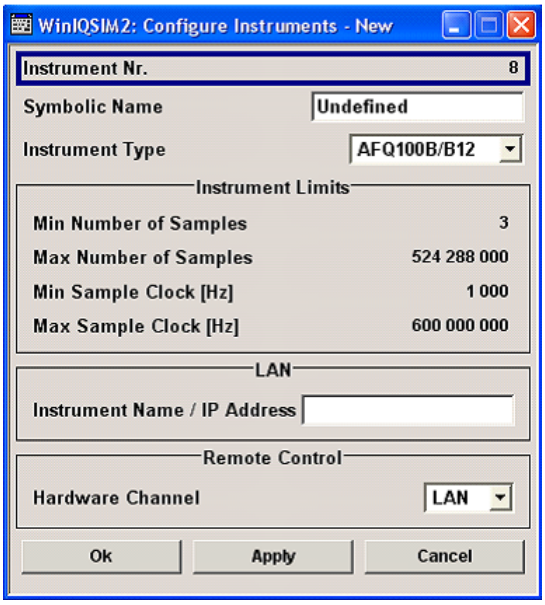
5.6.1 Creating New Instruments

1. In the Main window, select "Transmission > Instruments".

   The Configure Instruments dialog opens.

2. Click the "New" button.

   The "Configure Instruments - New" dialog opens to register a new instrument. The number of the instrument is counted by R&S WinIQSIM2 by default.



WinIQSIM2: Configure Instruments - New

Instrument Nr.	8
Symbolic Name	Undefined
Instrument Type	AFQ100B/B12

Instrument Limits

Min Number of Samples	3
Max Number of Samples	524 288 000
Min Sample Clock [Hz]	1 000
Max Sample Clock [Hz]	600 000 000

LAN

Instrument Name / IP Address

Remote Control

Hardware Channel: LAN

3. In the "Symbolic Name" field, enter an alias name of your choice.

4. In the "Instrument Type" field, select the device family to connect to.

   Tip: The option selected determines the maximum number of samples that can be generated.

5. In the "Instruments Limits" group box, important limits of the selected instrument are displayed.

6. If remote control connection via LAN interface is chosen, fill in the IP Address or instrument name for the connected instrument in the "Instrument Name/IP Address" field of the LAN group box.

7. In the "Hardware Channel" field of the "Remote Control" group box, select the VISA connection. For each of the channels, unique address information is required, the user interface changes accordingly.

Channel	Address name
USB	Serial Number, e.g., *100012* The number can be found on the rear panel of the instrument, above the barcode.
LAN	Instrument name / IP Address The instrument name consists of the following parts: *RS<instrument name><serial number>* You find the 6-digit serial number on the rear panel of the instrument. Example: *RSAFQ100A100004*
GPIB	Address, e.g., *19*

8. Click "OK" to confirm the settings and to close the dialog.
9. Click the "Apply" button to confirm the settings.
10. Click "Cancel" button to discard settings and to close the dialog.

5.6.2 Editing Instruments

1. In the "Available Instruments" group box, the instruments are listed.
2. Select the instrument to be edited and click the "Edit" button.

   The "Configure Instruments - Edit" dialog opens.
3. Change the settings and click OK.

   The settings are edited.

5.6.3 Scanning for Available Instruments

1. Open the "Configure Instruments" dialog by selecting "Transmission > Instruments".

   The "Configure Instruments" dialog opens.
2. Click the "Scan" button to scan the desired domain for the instrument.

   All instruments in the desired domain which are connected to the LAN or via GPIB or USB and being switched on are displayed in the"Available Instruments" list.
   R&S WinIQSIM2 obtains all information for connecting to the instrument, so further configuration is not necessary.

For further details regarding the prerequisites using the scan feature, refer to chapter 4.3.1, "Configuring an Instrument", on page 16.

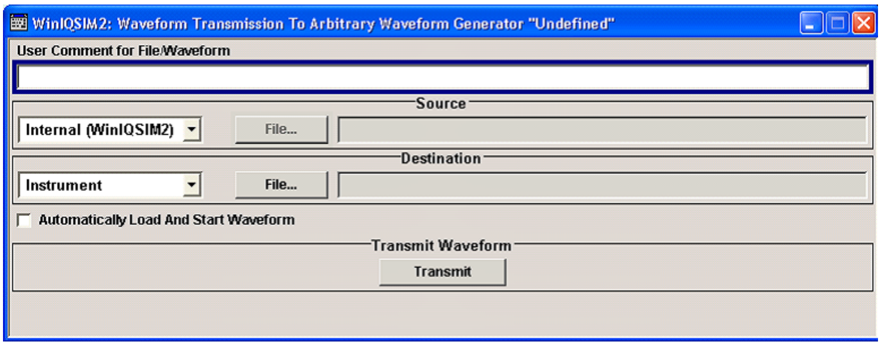
5.6.4 Deleting Instruments

1. In the Main window, select "Transmission > Instruments".
   The "Configure Instruments" dialog opens.
2. Select the instrument to be deleted and click the "Delete" button.

   The selected instrument is deleted.

5.6.5 Transmitting the Signals

1. In the Main window, select "Transmission > Transmit".
   The "Waveform Transmission To Arbitrary Signal Generator" dialog opens.


WinIQSIM2: Waveform Transmission To Arbitrary Waveform Generator "Undefined"
User Comment for File/Waveform
Source
Internal (WinIQSIM2)
File...
Destination
Instrument
File...
□ Automatically Load And Start Waveform
Transmit Waveform
Transmit

2. In the "User Comment for File/Waveform" field, additional information can be given, e.g., for identifying a file. The comment is written into the file and is shown in the "<Instrument> file open" dialog.
   Filling out this field is optional.
3. In the "Source" group box, there are two options:
   a) Internal
   The internal waveform file produced by the last signal generation is used as the transmission source. Because it is an internal file name, the file name of this file shown in the text field cannot be modified.
   b) File
   A file on the PC is used as the transmission source. You are prompted to choose an existing waveform file by the file browser. Select the file you want to transmit.
4. In the "Destination" group box, there are two options:
   a) Instrument
   Transmits the waveform to the instrument. Select "Instrument" and click the "File" button. You are prompted to select or to type in a file name from the instrument.
   Tip: In the instrument, the typed in name will be used and not the original name of the file.

b) File
Copies the waveform file to another file on the PC. Choose "File", and click the "File" button.
By clicking the "File" button, you are prompted to select or type in a file name on your hard disk. If you select a file as source and a file as destination, the transmission will be a simple copy operation.
If the "Automatically Load And Start Waveforms" check box is activated, the waveform file is copied to the hard disk of the instrument, loaded into the ARB, and started. The signal is immediately available at the I/Q output connectors.
If the checkbox is disabled, the waveform is copied from PC's hard disk to the instrument’s hard disk.

5. In the "Generate and Transmit Waveform" group box, click the "Transmit" button.

   The waveform is transmitted and saved according to the settings.

5.7 Graphics Menu

In the following, all settings concerning graphics are described. The output signal can be graphically displayed in a number of diagrams. This allows a fast check of signal characteristics. Zoom functions permits in-depth evaluation.

1. In the Main window, select "Graphics > Graphic 1 > Graphic 1 (Preview Only)".

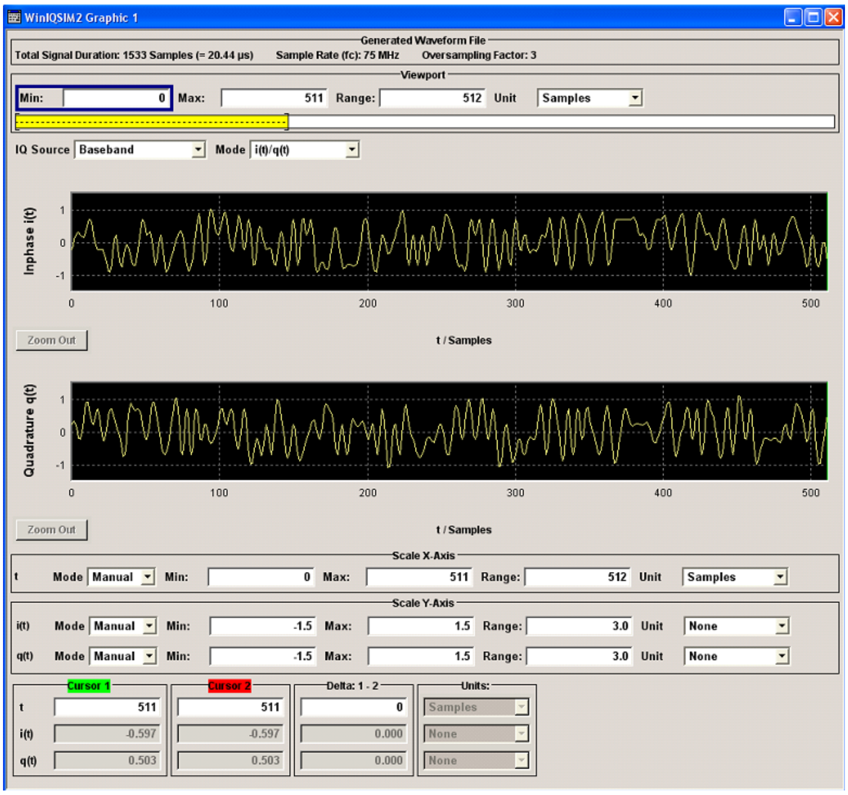
   Below the block diagrams, a small representation of the graphics dialog appears. A graphics icon in the signal flow indicates the signal source (either "Baseband", "AWGN", or "Baseband + AWGN").

2. Double-click the graphic.

   The Graphic 1 dialog opens.

5.7.1 Graphic 1-3

The graphical display of the waveform enables the user to rapidly check signal characteristics. Zoom functions allow in-depth evaluation.



WinIQSIM2 Graphic 1

Generated Waveform File

Total Signal Duration: 1533 Samples (= 20.44 µs) Sample Rate (fc): 75 MHz Oversampling Factor: 3

Viewport

Min: 0 Max: 511 Range: 512 Unit Samples

IQ Source Baseband Mode i(t)/q(t)

Scale X-Axis

t Mode Manual Min: 0 Max: 511 Range: 512 Unit Samples

Scale Y-Axis

i(t) Mode Manual Min: -1.5 Max: 1.5 Range: 3.0 Unit None

q(t) Mode Manual Min: -1.5 Max: 1.5 Range: 3.0 Unit None

	Cursor 1	Cursor 2	Delta: 1 - 2	Units:
t	511	511	0	Samples
i(t)	-0.597	-0.597	0.000	None
q(t)	0.503	0.503	0.000	None

The "Graphic" dialog consists of three main parts:

- Generated Waveform File
- Viewport selection
- Graphics area with scaling option and cursor settings (see chapter 5.7.4, "Graphical Representation of Waveforms", on page 38)

Since the behavior of these parts mostly depends on each other, the functions are explained without a strict separation into these parts.

5.7.2 Generated Waveform File

In the group box "Generated Waveform File", information on signal duration and sample rate is displayed.

Total Signal Duration
Displays the signal duration in seconds of the available samples within the generated waveform file.

SCPI command:
`GENerate:WAVeform:DURation` on page 194

Sample Rate (fc)
Dispalys the Sample Rate in [Hz], that is calculated as follow:

The Sampling Rate = Number of Symbols * Oversampling Factor.

SCPI command:

`GENerate:WAVeform:SAMPles` on page 195

`GENerate:WAVeform:SRATe` on page 195

Oversampling Factor

Displays the baseband oversampling factor.

This parameter is determined by the content of the generated waveform file. To adjust the oversampling rate, set the corresponding parameters defined within the standards.

SCPI command:

`GENerate:WAVeform:OSAMpling` on page 195

5.7.3 Viewport

In the following, the selection of the displayed area of the waveform takes place. Information about the total duration of the signal and the sample rate is given.

1. In the "IQ Source" field, select whether the signal is displayed before summation with AWGN, with AWGN only, or after the summation point (noise afflicted baseband signal).
2. To control the mode of representation, select it in the Mode combo box.

   The "Viewport" bar indicates the following information depending on the selected mode of representation:

"Mode"	**"Viewport"** 🟨	**Yellow Bar**	**White Bar**
"i(t)/q(t)" "r(t)/q(t)"	Scope, within the scaling of the X-axis can be shifted against the display.	Scope which is viewed	Total signal
"FTT"	Maximum observed scope	Actually used scope for FTT	Total signal
"Vector" "Constellation" "Eye" "CCDF"	Observed scope	Irrelevant	Total signal

Viewport

Note: There is no limitation to the length of the waveforms. Long waveform streams are divided in packets of an displayable size and are then displayed packet-wise.

In the "Viewport" group box, the following fields are displayed:

"Min"	Begin of scope.
"Max"	End of scope.

"Range" Delta between values in the "Min" and in the "Max" fields.
Tip: It also possible to set the "Min" value and the "Range" value, the "Max" value will be adjusted automatically.

"Unit" Units of the "Min", "Max", and "Range" values. If the unit is changed, the values contained in the "Min", "Max", and "Range" fields will be converted. Subsequently, the values in the new unit can be changed.

Mode
Controls the mode of representation.

For more information see chapter 5.7.4, "Graphical Representation of Waveforms", on page 38.

"i(t)/q(t)" Displays the actual waveform over time in two windows: the top one for inphase, the bottom one for quadrature samples.

"r(t)/phi(t)" Displays the actual waveform over time in two windows: the top one for amplitude, the bottom one for phase values.

"Vector" The samples are depicted in the complex plane, transitions displayed as vectors.

"CCDF" Complementary Cumulative Density Function

"FFT Magnitude" Spectrum (magnitude over frequency) derived from a Fast Fourier Transform of the signal.

"Constellation" All sample points displayed in the complex plane.

"Eye(i/q)" Eye diagram of I and Q samples. Superposition in time of symbol transitions sampled at symbol rate.

5.7.4 Graphical Representation of Waveforms

There are two ways of zooming into a representation for measurement:

- By mouse
- By value entry

Zooming into a representation by mouse

1. To zoom into a displayed part of a waveform, position the mouse cursor in the top left-hand corner of the graph area to be zoomed.
2. Click the left mouse key and hold it down.

   The mouse cursor will change its shape to a cross-hair.
3. Move the mouse cursor to the bottom right-hand corner of the section to be zoomed. While you move the cursor, a dotted rectangular frame will be drawn between the start point and the current cursor position. The frame marks the area to be zoomed.
4. When the mouse has arrived at the end of the area to be zoomed, i.e. when the whole area is within the red frame, release the left mouse key.

   The selected area is shown expanded.

Zooming into a representation by value entry

Directly beneath the graphics area, the scaling section with "Scale X-Axis" group box and "Scale Y-Axis" group box is located. The behavior for X- and Y-axis is identical. For that reason, the function is explained only once.

► In the "Scale X-Axis" group box, in the "Mode" combo box, there are two options for scaling of modes:

  a) "Auto"
  If "Auto" is selected, the scaling works automatically and all entry fields are inactive. Only the unit of the values which will adapt in the gray entry fields can be selected. The automatic scaling for the X-axis takes the values of the Viewport; the Y-axis scaling is calculated from the maximum and minimum value of the samples displayed.

  b) "Manual"
  If "Manual" is selected, the scaling is fully adjustable by the user. The area that is to be zoomed is set by the "Min" and "Max" values. The "Range" value is then adapted. As in "Viewport", you can as well set the "Min" and the "Range" value. The "Max" value is adapted then.
  By means of the "Unit" combo box, the unit of the values can be changed again, the values are converted automatically.

  Tip: The scaling values can never be higher or lower than the scaling values set under "Viewport".

Setting graphics cursors into a representation

There are two cursors available for measurements inside the graphs: A red cursor and a green one. The cursors are shaped as a vertical line. The cursors can be placed by mouse in certain positions.

1. Left-click on one of the cursors and hold down the left mouse button. Drag the mouse to the position where you want to place the cursor. Release the mouse button.

   Note: If you choose a "Viewport" where the cursors are not located, it will be impossible to use the mouse to manipulate the cursors. In this case or in order to place the cursors precisely, the cursor can be positioned by an value entry.

2. In the "Cursor 1" group box, the cursor position on the X-axis is set. In the "Cursor 2" group box, the cursor position on the Y-axis is set. Type the exact position in the "t" field.
   The cursor moves to the defined position, giving the Y-axis values in the "i(t)" und "q(t)" fields. The values are not editable. In the "Delta: 0 - X" group box, the difference (delta) between the X-axis value and Y-axis value of the cursor position is displayed.

3. Optional, in the "Delta: 1 - 2" group box, a value can be entered and the red cursor will adapt to the desired delta value.

   The signal can be analyzed.

Using the Zoom Out button

► If a waveform has been zoomed, the "Zoom Out" button changes from inactive to active.

Tip: The "Zoom Ou"t button always returns the display to the state before all zoom operations.

5.8 Help Menu

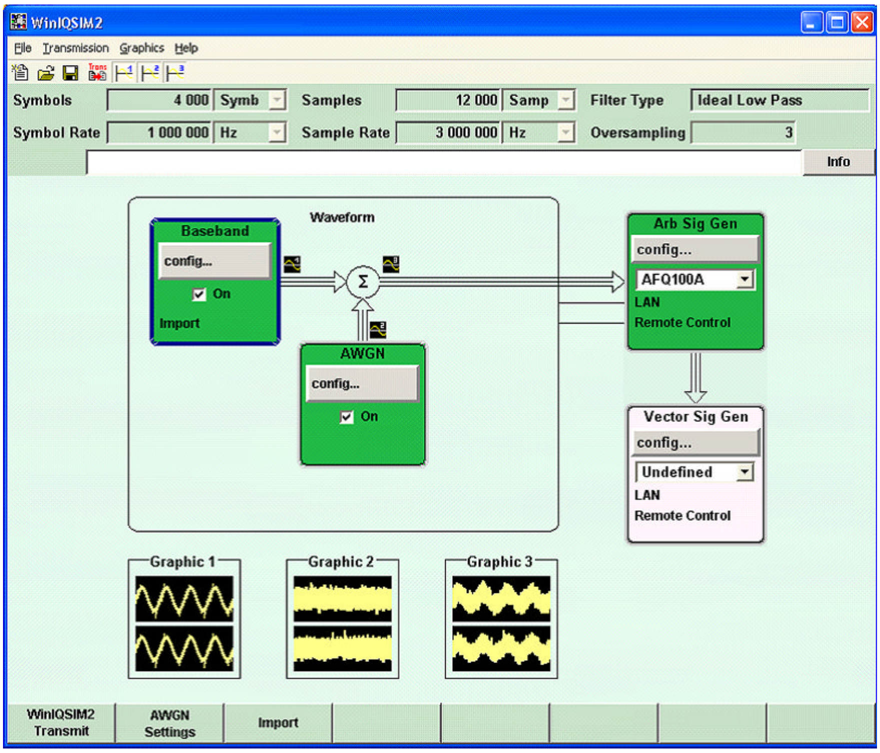
R&S WinIQSIM2 is equipped with a context-sensitive help function. A help page can be called any time during software operation. The context-sensitive page which is opened with the F1 button is part of a comprehensive help system.

It is possible to move from this context-sensitive page to any page of the help system. An overview of the contents of the online help can be reached via the menu "Help > Contents".

A search for keywords within the help function is available via menu item "Help > Index".

5.9 Main Window - Signal Flow Diagram

The block diagram's look-and-feel is based on the R&S SMU firmware.



Symbols	4 000 Symb	Samples	12 000 Samp	Filter Type	Ideal Low Pass
Symbol Rate	1 000 000 Hz	Sample Rate	3 000 000 Hz	Oversampling	3

Arb Sig Gen: AFQ100A

Vector Sig Gen: Undefined

The signal flow diagram is the core control element of R&S WinIQSIM2. The framed area symbolizes R&S WinIQSIM2 containing the blocks for signal generation and signal alteration. The block diagram shows provided options, signal configuration, and the signal flow of the generator. Signal generation can be completely operated from the block diagram. The highlighted function block can be directly switched on and off with the On / Off check box.

The Main window of R&S WinIQSIM2 contains the whole system: The PC running R&S WinIQSIM2 displayed as one entity in the frame as well as the connected instruments. The user controls the whole system from one panel.

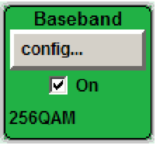
The signal flow is represented by thick arrows. Summation points are indicated by a sum sigma. Control connections are displayed as thin lines leading from R&S WinIQSIM2 to the corresponding blocks that represent external instruments. Invalid connections are symbolized by crossed out lines.

Each block represents a function of signal generation. The function is indicated in the headline of the block, e.g., "Baseband". In this block, digital standards, custom digital modulation, and multi carrier scenarios are set, for instance. In the "AWGN" block, additive white gaussian noise can be added to the baseband signal.

Outside the frame, the signal or modulation generators that are connected or can be connected are symbolized by two blocks; "Arb Sig Gen" represents the ARB Modulation Generators and "Vec Sig Gen" represents the Vector Signal Generators. Two connection lines symbolize the two possible connections ("LAN" or "Remote Control") between the PC with WinIQSIM2 and the instrument.

5.9.1 Working with the Baseband Block

For a detailed description of the settings provided in the "Baseband" block, refer to the corresponding section in chapter 5.10, "Baseband Block", on page 48.



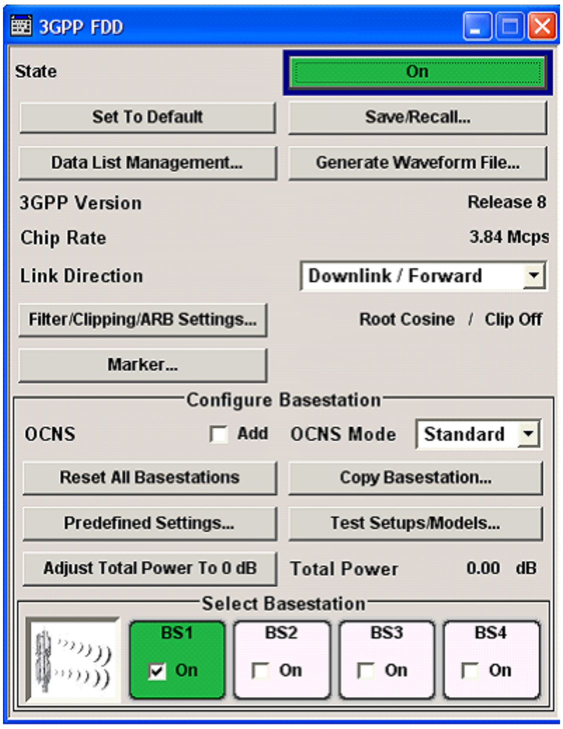
1. In the "Baseband" block diagram, below the "Config" button, the state check box is located. Here, the states On or Off of the component block are controlled. By activating the block, it turns its color to green and contributes to the signal flow from now on. Click the "Config" button.

   A menu with a list of optional standards opens.
   The available digital standards depend on the installed options.

2. Select one of the offered options.

   A dialog opens. Parameter entries for the selected digital standard can be made.

   Example: Standard "3GPP FDD"



State: On

3GPP Version: Release 8

Chip Rate: 3.84 Mcps

Link Direction: Downlink / Forward

Root Cosine / Clip Off

OCNS □ Add OCNS Mode: Standard

Total Power 0.00 dB

BS1 ☑ On BS2 □ On BS3 □ On BS4 □ On

Tip: For more information on generation of digital standard waveforms, see the context-sensitive help page of the corresponding dialog (opened with the F1 button) or the help system.
An overview of the contents of the online help can be reached via the menu "Help > Contents".

Main difference between signal generation in real time and with the signal generation software

Some of the hardware specific components of the instruments are not relevant for R&S WinIQSIM2, because R&S WinIQSIM2 is not based on real time hardware. For this reason, the following features are not supported:

- Trigger
- Clock
- Marker delay

In the "Filter" settings dialog, the R&S WinIQSIM2 provides the following additional parameters:

- "Impulse length"
  Displays the number of filter tabs. If the check box is activated, the most sensible parameter values are selected. The value depends on the coherence check. If the check box is deactivated, the values can be changed manually.
- "Oversampling"
  Determines the upsampling factor. If the check box is activated, the most sensible parameter values are selected. The value depends on the coherence check. If the check box is deactivated, the values can be changed manually.

When the digital standard is activated, a coherence check is performed. The entries are checked.

The following parameters are part of the check:

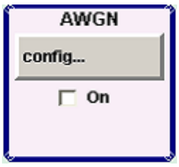
- "Sequence Length"
- "Impulse Length"
- "Oversampling"
- instrument limits.

When the coherence check is positive, the waveform is generated and the internally used parameter values are displayed in the gray fields.

5.9.2 Working with the AWGN Block

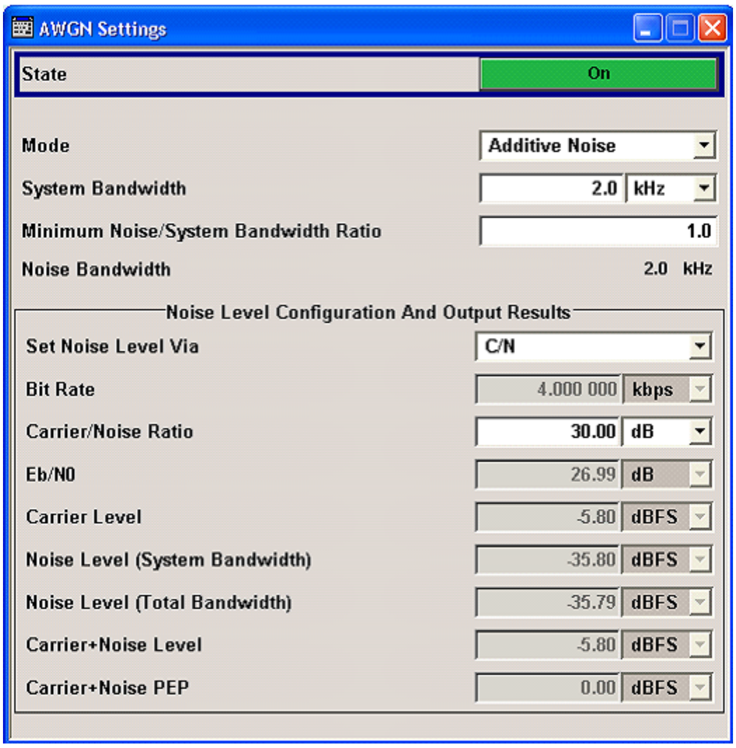
For a detailed description of the settings provided in the AWGN block, refer to chapter 5.11, "AWGN - Noise Generator", on page 121.

Adding AWGN to the Baseband Signal

1. 

   In the "AWGN" block diagram, click "Config".

   The "AWGN Settings" dialog opens.



AWGN Settings

State	On	
Mode	Additive Noise	
System Bandwidth	2.0	kHz
Minimum Noise/System Bandwidth Ratio	1.0	
Noise Bandwidth	2.0	kHz

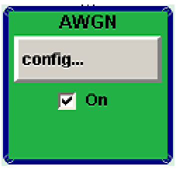
Noise Level Configuration And Output Results

Set Noise Level Via	C/N	
Bit Rate	4.000 000	kbps
Carrier/Noise Ratio	30.00	dB
Eb/N0	26.99	dB
Carrier Level	-5.80	dBFS
Noise Level (System Bandwidth)	-35.80	dBFS
Noise Level (Total Bandwidth)	-35.79	dBFS
Carrier+Noise Level	-5.80	dBFS
Carrier+Noise PEP	0.00	dBFS

2. In the"AWGN Settings" dialog, make the desired settings.

3. Click the "Off" button to change the AWGN state from Off to On.

   Tip: For multi carrier and multi segment signals, the AWGN status is set to Off.

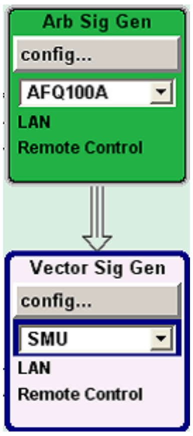
   The AWGN is activated and added to the baseband signal. The active AWGN is indicated in the AWGN block diagram.

5.9.3 Working with the Gen Blocks Arb Sig and Vec Sig

The block diagrams symbolize the signal or modulation generators that are connected or can be connected to R&S WinIQSIM2.

- "Arb Sig Gen" represents an ARB Modulation Generators, e.g., R&S AFQ.
- "Vec Sig Gen" represents Vector Signal Generators, e.g., R&S SMU.

The two blocks are operated in a similar fashion so both blocks are referred to as "Sig Gen" block.



Selecting an Instrument in a Sig Gen Block

► Select "Main window > Sig Gen block" and select an instrument form the combo box.

The list contains the alias names of the defined instruments.

In the "ARB Sig Gen" block, only ARB instruments are listed, in the "Vec Sig Gen" block only vector signal generators are listed.

For more information see section chapter 5.6.1, "Creating New Instruments", on page 32.

Two connection lines symbolize the two possible physical connections between instrument and the PC with R&S WinIQSIM2. The lines are labeled with "LAN" and "Remote Control" (remote control connection via VISA). If SCPI via LAN is used, only one physical line is indicated.

If the instrument is not responding on the corresponding connection, the line will be crossed out. The instrument is offline.

For signal calculation, it is not necessary that an instrument is connected. The signal calculation can be performed for a dummy instrument (offline) and the waveform can later be transmitted to an instrument (online). However, keep in mind, that the general instrument settings affect various functions, e.g., the maximum size of the file the waveform is stored in. Therefore, it is important that the calculation is performed for exactly the instrument which will later be the destination for the signal.

Configuring Instrument Related Issues

► To configure the signal generation, in the "Sig Gen Block", click the "Config" button.

A menu opens and the following options for configuring the signal generation are displayed:

"Instruments"	Opens the "Instruments" dialog. For more information see chapter 5.6.1, "Creating New Instruments", on page 32.
"Remote Desktop"	Opens the standard Windows Remote Desktop connection to the chosen instrument.
"Remote Control (SCPI)"	Opens the "Remote Control" menu. For more information see chapter 5.9.4, "Remote Control Dialog", on page 46.

If the Remote Desktop connection or the Remote Control dialog is active, the block turns green to indicate an active remote connection to the instrument.

5.9.4 Remote Control Dialog

This dialog provides the option to remotely control the chosen instrument by means of a basic set of control functions. The instrument can be controlled without Remote Desktop.

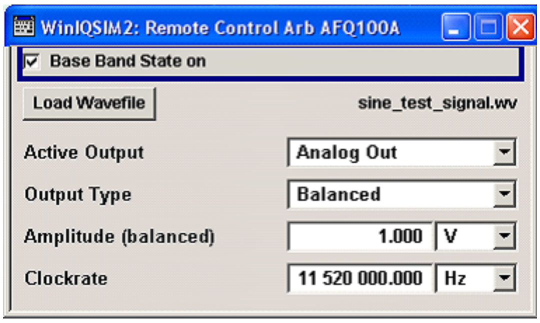
In the Remote Control dialog, the current instrument configuration is always displayed and can be checked. Here, the settings of the instrument can be changed remotely.

If one or more of these settings are changed on the instrument without using the Remote Control menu, the menu will be updated automatically to the new settings.

When the "Remote Control" dialog is open, the "Remote LED" on the Front Panel of the instrument starts to flash, indicating that the instrument is set to remote state periodically. This is necessary to query the instrument status. The flashing stops when the "Remote Control" dialog is closed.

Remote Control of the ARB Generator

► In the Main window, select "Arb Sig Gen Block > Config > Remote Control (SCPI)".


WinIQSIM2: Remote Control Arb AFQ100A

☑ Base Band State on

Load Wavefile sine_test_signal.wv

Active Output: Analog Out

Output Type: Balanced

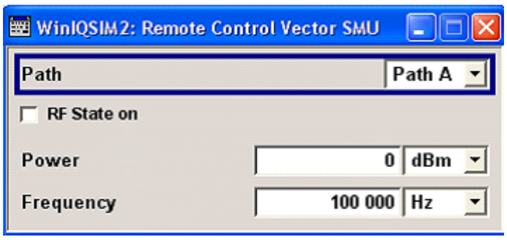
Amplitude (balanced): 1.000 V

Clockrate: 11 520 000.000 Hz

The Remote Control ARB dialog opens.
The display of the dialog depends on the selected instrument.

Remote Control of the Vector Signal Generator

► In the Main window, in the "Vec Sig Gen Block > Config > Remote Control (SCPI)".



WinIQSIM2: Remote Control Vector SMU

Path: Path A

□ RF State on

Power: 0 dBm

Frequency: 100 000 Hz

The Remote Control Vec Sig Gen dialog opens.

Remote Control ARB

In the "Remote Control ARB" dialog, the following fields and buttons are displayed:

"Base Band State on" Enables/disables the baseband output of the instrument.

"Load Wavefile" Opens the SCPI File Explorer to select a file on the instrument which is to be loaded into the ARB. The file is displayed in the area right of the button.

"Active Output" Sets the active output of the instrument to Analog out or Digital out, if the instrument is equipped accordingly.

"Output Type" Selects analog output type options:

- Balanced
- Unbalanced
- No options for digital output type.

"Amplitude" Sets the amplitude of an instrument. This depends on the settings of "Active Output" and "Output Type". The analog output amplitudes are given in Volt, the digital output amplitudes in a value related to Full Scale (FS).

"Clockrate" Sets the clock rate of the instrument.

Remote Control Vec Sig Gen

In the "Remote Control Vec Sig Gen" dialog, the following fields are displayed:

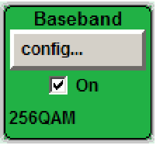
"Path" Selects the path to be controlled if the instrument is equipped with more than one path.

"RF State on" Enables/disables the radio frequency output of the instrument.

"Power" Sets the power of the instrument.

"Frequency" Sets the frequency of the instrument.

5.10 Baseband Block



The Signal Generation Software R&S WinIQSIM2 provides you with the ability to generate digital modulation signals in accordance with the definitions in the digital standards or with user-definable characteristics. You can then load these files into the R&S Signal Generator in the form of waveform files.

Several digital standards are provided, plus digital modulation with user-definable signal characteristics and generation of multi carrier CW signals with the aid of Multi Carrier CW.

The settings for digital modulation can be accessed in the block diagram via the Baseband function block.

The equipment layout for generating the internal, digital modulation signals includes the options Multi Carrier CW Signal Generation and the option Import IQ Data.

In addition, the appropriate option is required for the digital standards. These options are specified in the description of the respective standard.

In the subsequent signal route the baseband signals can be loaded with noise (see chapter 5.9.2, "Working with the AWGN Block", on page 43).

5.10.1 Data and Signal Sources in Baseband

This section describes the common characteristics of the signals used for generating the baseband signal for all standards, including for example all listed data sources. The selection in the digital menus at any given time depends on the parameter and standard concerned and is clear from the selection list offered in the menu. The external data sources may therefore not be available in certain cases.

Characteristics which are uniquely specific to particular standards are described in the corresponding manual.

The following input signals are used when digital modulation signals are being generated:

- Modulation data
- Control signals

Externally computed waveform files can be loaded in the instrument using computer interfaces or IEC bus.

Internally the "Multi Carrier CW" menu for defining multi carrier waveforms is available (see chapter 5.10.3, "Multi Carrier Continuous Wave", on page 83).

5.10.1.1 Internal PRBS Data and Data Patterns

PRBS generators deliver pseudo-random binary sequences of differing length and duration. They are known as maximum length sequences, and are generated with the aid of ring shift registers with feedback points determined by the polynomial.

By way of example, the diagram below shows a 9-bit generator with feedback to registers 4 and 0 (output).

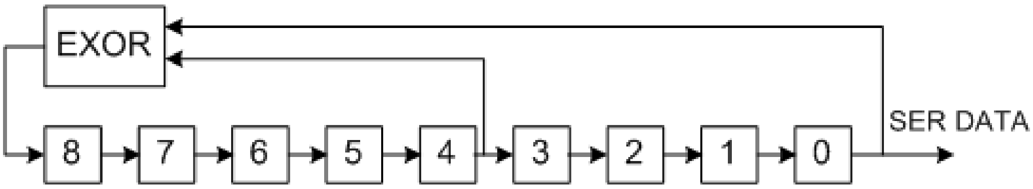
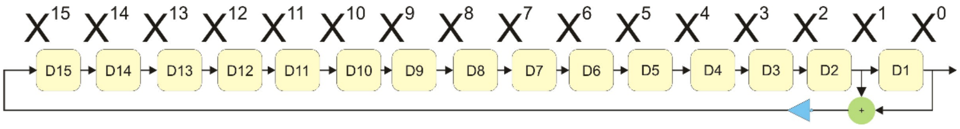


Fig. 5-1: 9-bit PRBS generator

For PRBS15 and PRBS23, a CCITT V.52-compliant data inversion is performed in the feedback path automatically as shown below:



The pseudo-random sequence from a PRBS generator is uniquely defined by the register number and the feedback. The following table describes all the available PRBS generators:

Table 5-3: PRBS generators

PRBS generator	Length in bits	Feedback to	Menu selection
9-bit	29 -1 = 511	Registers 4, 0	PRBS 9
11-bit	211 -1 = 2047	Registers 2, 0	PRBS 11
16-bit	216 -1 = 65535	Registers 5, 3, 2, 0	PRBS 16
20-bit	220 -1 = 1048575	Registers 3, 0	PRBS 20
21-bit	221 -1 = 2097151	Registers 2, 0	PRBS 21

The "Data Source" selection for PRBS data from the menus is "PN11, ...".

Simple data patterns such as binary 0 strings or 1 strings can also be used as internal modulation data. The "Data Source" selection from the menus is "ALL 0", "ALL1". A variable bit string with a maximum length of 64 bits can be entered in an input field by selecting "Pattern."

5.10.1.2 Internal Modulation Data from Lists

The file name can be defined by the user.

The "Data Source" selection from the menus is "Data List". When this menu item is chosen the "Select Data List" button appears; this opens the file-selection window for data lists. A list can be created and edited in the "Data List Editor", which is accessed from the menu via the "List Management" or "Data List Management" button.

The maximum length of a data list is determined by the size of the data list memory (see data sheet). There is no restriction on the number of lists that can be stored.

5.10.1.3 Control Signals

The following control signals are processed in the R&S WinIQSIM2:

- "Burst Gate" for power ramping
- "Level Attenuation" for power ramping

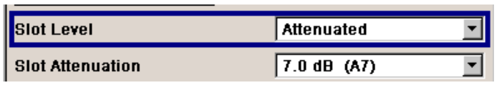
"CW" for controlling the CW (continuous wave) mode is not used in R&S WinIQSIM2. However, a control list generated for R&S Signal Generator can contain CW controls.

A separate file with the file extension `*.dm_iqc` is created for each defined control signal.

Power Ramping and Level Attenuation

When power ramping is enabled, a ramp is generated whenever there is a data switch on the "Burst" signal (from high to low or low to high). The steepness of this ramp can be adjusted. Power ramping is enabled and configured in the "Power Ramp Control" submenu.

The "Lev_Att" signal is used to control a defined level attenuation. If level attenuation is enabled, the modulation signal level is attenuated by a defined value if the "Lev_Att" signal is high. The level attenuation value is defined in the "Power Ramp Control" menu. For the "GSM/EDGE" standard a maximum of 7 different level attenuation values can be defined and allocated separately to the 8 slots quite independently of one another. Level attenuation is enabled either in the "Power Ramp Control" menu (Custom Digital Modulation) or in the "Burst Editor" (GSM/EDGE).



Slot Level: Attenuated

Slot Attenuation: 7.0 dB (A7)

Level attenuation enables to simulate radio stations located at various distances.

The diagram below shows an example of how the power ramping signals work.

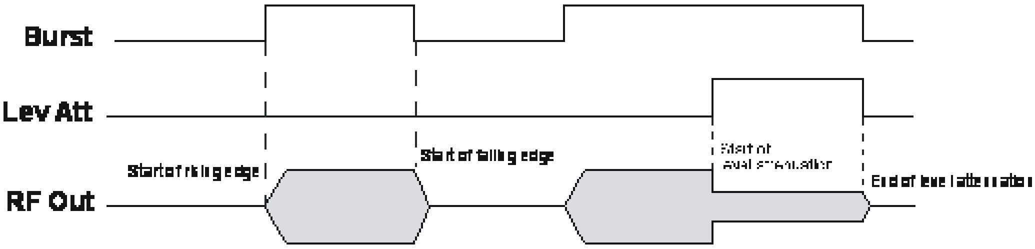


Fig. 5-2: Signal behavior when power ramping is enabled. The Burst Gate signal defines the start of the rising and falling edges of the envelope, and the Lev Att signal defines the start and end of level attenuation. The level attenuation value is defined in the Power Ramp Control menu.

5.10.1.4 Marker Output Signals

The R&S WinIQSIM2 generates user-definable marker information which will can be processed by the R&S Signal Generator to output the appropriate markers signals.

All four available marker outputs are configurable.

5.10.2 Digital Modulation - Custom Digital Modulation

The R&S Signal Generator can generate digital modulation signals with user-definable characteristics. Baseband filtering and the symbol rate can be set within wide limits.

5.10.2.1 Modulation Types - Custom Digital Mod

The available predefined modulation types are ASK (amplitude shift keying), FSK (frequency shift keying), PSK (phase shift keying) and QAM (quadrature amplitude modulation). Additionally, a user defined modulation mapping can be applied (see "User Mapping" on page 54).

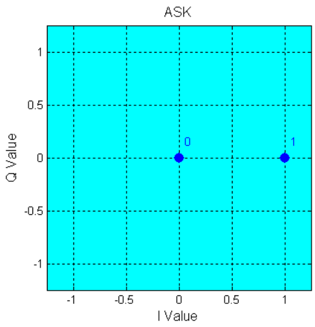
The actual modulation procedure is described by mapping, which assigns I and Q values (PSK and QAM) or frequency shifts (FSK) to every modulation symbol that occurs. This is represented graphically in the constellation diagrams.

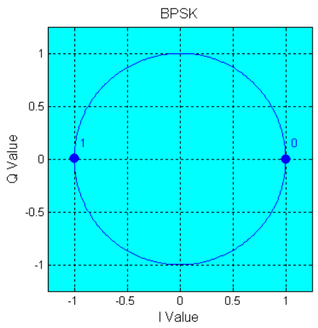
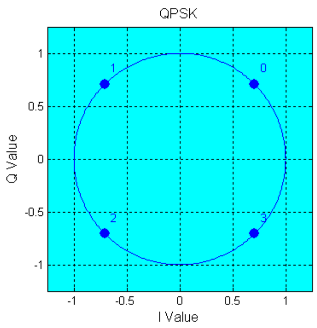
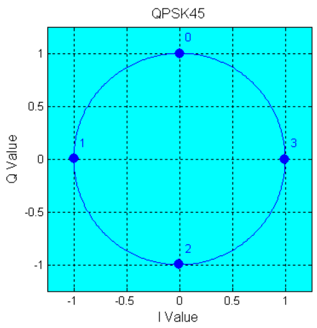
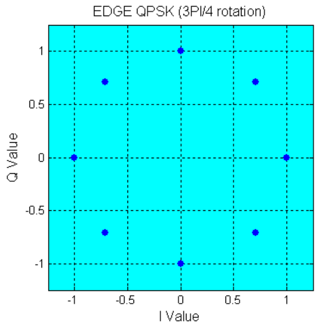
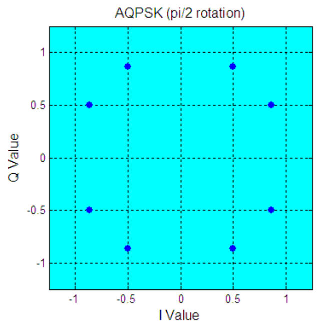
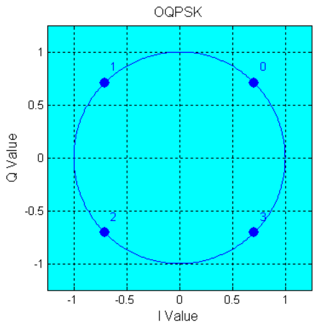
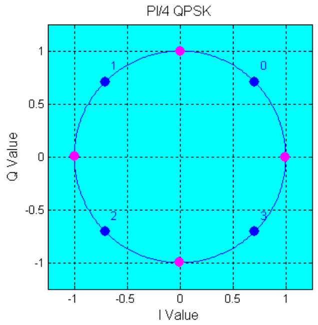
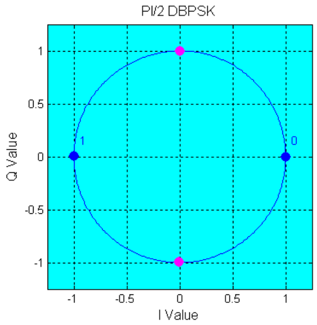
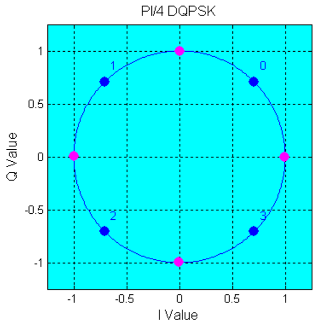
The mapping for the selected modulation type is displayed in the "More..." submenu in the "Modulation" menu section (see "More - Modulation Type - Digital Modulation" on page 69).

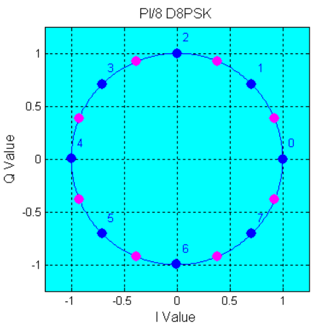
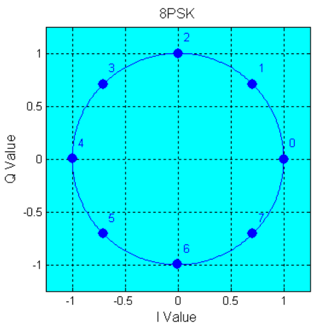
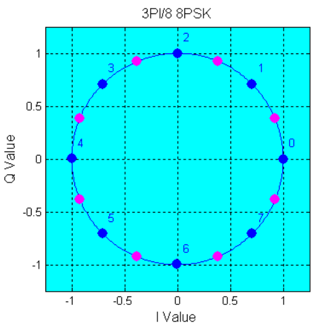
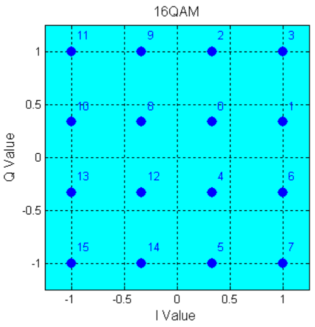
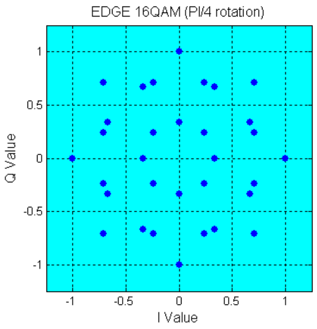
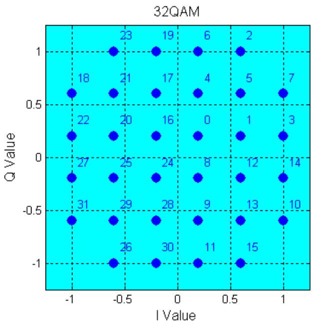
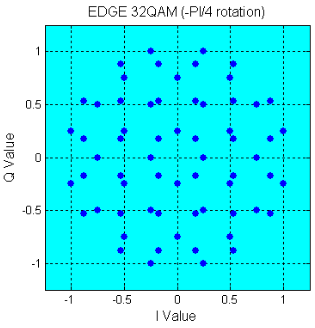
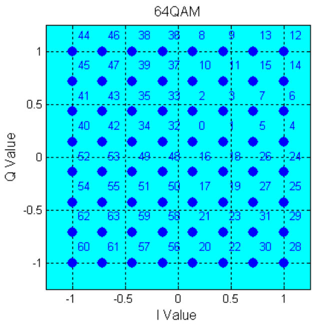
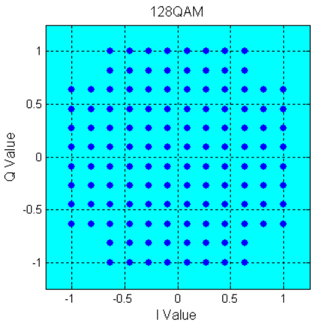
The QAM procedures 16QAM, 32QAM, 64QAM have been produced in accordance with ETSI standard ETS 300429 for digital video broadcasting (DVB). The QAM procedures 256QAM and 1024QAM are not specified in this standard, but have been produced according to the same basic principles.

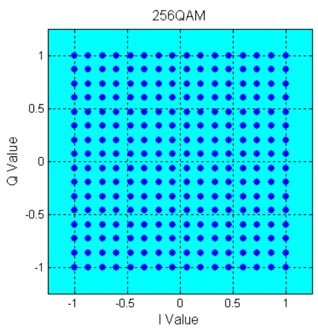
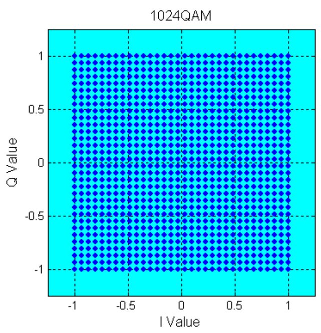
In the case of all FSK procedures, the user can set the symbol rate fSYMB up to a maximum value (see data sheet). If MSK is selected, the frequency deviation (FSK deviation) cannot be set since it is permanently set to ¼ of the symbol rate.

Table 5-4: Modulation type and associated mapping

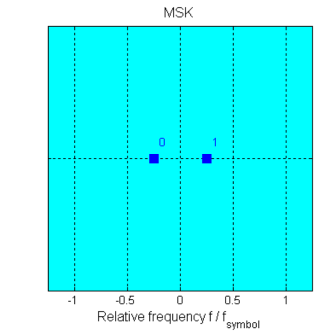
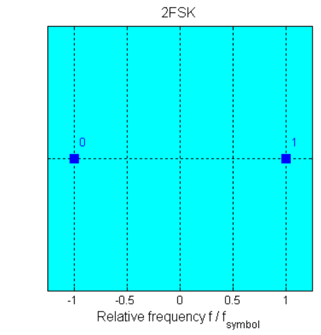
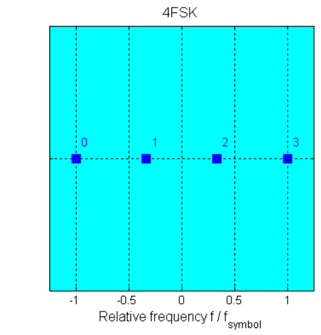
ASK 1 bit per symbol `SOUR:BB:DM:FORM ASK`		
		
PSK		
BPSK 1 bit per symbol SOUR:BB:DM:FORM BPSK	QPSK 2 bits per symbol `SOUR:BB:DM:FORM QPSK`	QPSK 45° Offset 2 bits per symbol, 45° rotation `SOUR:BB:DM:FORM QPSK45`

		
QPSK EDGE 2 bits per symbol, 3pi/4 rotation `SOUR:BB:DM:FORM QEDGe`	AQPSK 2 bits per symbol, pi/2 rotation `SOUR:BB:DM:FORM AQPSk`	OQPSK 2 bits per symbol, Q offset `SOUR:BB:DM:FORM OQPSK`
		
pi/4-QPSK 2 bits per symbol pi/4rotation `SOUR:BB:DM:FORM P4QP`	pi/2-DBPSK 1 bit per symbol Differential coding, pi/2 rotation `SOUR:BB:DM:FORM P2DB`	pi/4-DQPSK 2 bits per symbol Differential coding, pi/4 rotation `SOUR:BB:DM:FORM P4DQ`
		
pi/8-D8PSK 3 bits per symbol Differential coding, pi/8 rotation `SOUR:BB:DM:FORM P8D8`	8PSK 3 bits per symbol `SOUR:BB:DM:FORM PSK8`	8PSK EDGE (3pi/8 8PSK) 3 bits per symbol Edge coding, 3pi/8 rotation `SOUR:BB:DM:FORM P8ED`

		
QAM		
16QAM 4 bits per symbol `SOUR:BB:DM:FORM QAM16`	16QAM EDGE 4 bits per symbol, pi/4 rotation `SOUR:BB:DM:FORM QAM16EDge`	32QAM 5 bits per symbol `SOUR:BB:DM:FORM QAM32`
		
32QAM EDGE 5 bits per symbol, -pi/4 rotation `SOUR:BB:DM:FORM QAM32EDge`	64QAM 6 bits per symbol `SOUR:BB:DM:FORM QAM64`	128QAM 7 bits per symbol `SOUR:BB:DM:FORM QAM128`
		
256QAM 8 bits per symbol `SOUR:BB:DM:FORM QAM256`	1024QAM 10 bits per symbol `SOUR:BB:DM:FORM QAM1024`	

		
FSK		

MSK
1 bit per symbol, FSK deviation
`SOUR:BB:DM:FORM MSK`
`SOUR:BB:DM:FSK:DEV 0.1 MHz`
	2FSK
	1 bit per symbol
	`SOUR:BB:DM:FORM FSK2`
		4FSK
		2 bit per symbol
		`SOUR:BB:DM:FORM FSK4`

In addition to the FSK modulations, a variable FSK modulation with definable deviation per symbol is available.

`[:SOURce<hw>]:BB:DM:FORMat` on page 241

User Mapping

A user defined modulation mapping file can also be selected as modulation mapping source. The user modulation mapping file must have extension `*.vam` and can be created with the R&S mapping wizard. The mapping wizard (mapwiz) is a tool from Rohde & Schwarz designed for editing modulation schemes (e.g. QPSK, 32QAM). Its main purpose is the assignment of logical symbol numbers to constellation points and the selection of modulation specific parameters. Beyond this it supports the creation of nearly any arbitrarily chosen constellation diagram. The output of mapwiz is a mapping file (`*.vam`) that can be imported on a R&S WinIQSIM2. The program was developed on a 32-bit Microsoft Windows platform under MATLAB. For more information, refer to the description "Introduction to "mapwiz" Mapping Editor" on the Rohde&Schwarz Internet page.

5.10.2.2 Coding - Custom Digital Mod

Modulation symbols are coded directly before I and Q values or frequency shifts are assigned. Coding is thus directly related to modulation methods, which is the reason why codings are not freely combinable with modulation methods. The following table shows which of the coding combinations are available and defines the modulation types for which the various coding procedures can be used.

In the notation used below an denotes the n-th input symbol and bn denotes the correspondingly coded output symbol. Individual bits in the symbols from the LSB (least significant bit) to the MSB (most significant bit) are denoted by a0n, a1n and so on. The same applies to the output symbols.

Table 5-5: Permissible coding combinations for modulation symbols and modulation type

	OFF	Differential	Phase Diff.	Diff. +Gray	Gray	GSM	NADC, PDC, PHS, TETRA, APCO25 (PSK), PWT	TFTS/ TETRA	INMARSAT, ICO, WCDMA, cdma2000	VDL	APCO25 (FSK)	APCO25 (8PSK)
ASK	X	X		X	X							
BPSK	X	X		X	X							
pi/2 DBPSK	X				X							
QPSK	X	X		X	X				X			
QPSK EDGE	X											
QPSK45° Offset	X	X		X	X							
Pi/4 QPSK	X	X			X							
Pi/4 DQPSK	X				X		X	X				
8PSK	X	X		X	X					X		
AQPSK	X	X		X	X							
OQPSK	X	X		X	X							
8PSK_EDGE	X											
Pi/8 D8PSK	X				X							X
MSK	X	X		X	X	X						
2FSK	X	X		X	X	X						
4FSK	X	X		X	X						X	
8FSK	X	X		X	X							
16FSK	X	X		X	X							
16QAM	X	X	X	X	X							
16QAM EDGE	X											
32QAM	X	X	X	X	X							

	OFF	Differ-ential	Phase Diff.	Diff. +Gray	Gray	GSM	NADC, PDC, PHS, TETRA, APCO25 (PSK), PWT	TFTS/ TETRA	INMARSAT, ICO, WCDMA, cdma2000	VDL	APCO25 (FSK)	APCO25 (8PSK)
32QAM EDGE	X											
64QAM	X	X	X	X	X							
128QAM	X	X	X	X	X							
256QAM	X	X	X	X	X							
1024QAM	X	X	X	X	X							

Modulation type pi/4DQPSK

With differential coding switched on at the same time, a constellation diagram is obtained for pi/4DQPSK which is similar to that obtained for 8PSK. Phase shifts are however assigned to the individual modulation symbols. The following tables show the assignment of modulation symbols to phase shifts of the I/Q vector in relation to the selected coding.

Table 5-6: Phase shifts for pi/4DQPSK

Modulation symbol a_n (binary indication: MSB, LSB)	00	01	10	11
Phase shifts without coding	+ 45°	+ 135°	- 135°	- 45°
Phase shifts with coding NADC, PDC, PHS, TETRA or APCO25 (PSK)	+ 45°	+ 135°	- 45°	- 135°
Phase shifts with coding TFTS	- 135°	+ 135°	- 45°	+ 45°

Coding algorithms

Common coding types are listed in the following table.

Table 5-7: Coding algorithms

Coding	Coding algorithm	Applicable for K bit/symbol
NONE	$b_n = a_n$	k = 1...8
Differential	$b_n = (a_n + b_{n-1})$ modulo 2^k	k = 1...7
Differential + Gray	Gray coding with additional differential coding	k = 1...7
GSM	dc_n = not (d_n exor d_{n-1})	k = 1

Example: Differential coding for QPSK modulation with K = 2 bit/symbol

Decimal display; value range for modulation symbols

$a_n \in \{0; 1; 2; 3\}$

Recursive coding is defined as follows: $b_n = (a_n + b_{n-1})$ modulo 4

Depending on the state of a preceding modulation symbol b_{n-1} the coded modulation symbol b_n is obtained for example from modulation symbol $a_n = 2$ as follows:

a_n = 2	**b_{n-1}**	**b_n**
	0	2
	1	3
	2	0
	3	1

By means of differential coding, the assignment between modulation symbols and phase differences shown in the following table is generated:

Modulation symbol a_n (binary, MSB, LSB)	00	01	10	11
Phase difference	0°	90°	180°	270°

Example: Gray and differential coding for 8PSK modulation

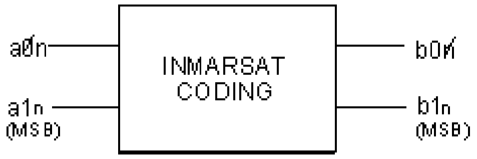
First, a gray coding is performed according to the gray code. Afterwards, a differential coding is performed according to the recursive coding algorithm quoted above. The assignment between modulation symbols and phase differences shown in the following table is generated:

"Differential coding according to VDL" can be used for modulation types with 3 bits/symbol, e.g. 8PSK.

Table 5-8: Differential coding according to VDL

Modulation symbol d_n (binary, MSB, LSB)	000	001	010	011	100	101	110	111
Phase difference	0°	45°	135°	90°	315°	270°	180°	225°

"Phase differential coding" INMARSAT and PHASE DIFF correspond to system standards Inmarsat-M and DVB according to ETS 300 429. The INMARSAT coding can generally be used for modulation types with 2 bits/symbol, such as QPSK. It uses the following algorithm.



$$\begin{aligned} b_{1n} = {} & [\text{NOT } (a_{1n} \text{ EXOR } a_{0n}) \text{ AND } (a_{1n} \text{ EXOR } b_{1(n-1)})] \text{ OR} \\ & [(a_{1n} \text{ EXOR } a_{0n}) \text{ AND } (a_{1n} \text{ EXOR } b_{0(n-1)})] \\ b_{0n} = {} & [\text{NOT } (a_{1n} \text{ EXOR } a_{0n}) \text{ AND } (a_{0n} \text{ EXOR } b_{0(n-1)})] \text{ OR} \\ & [(a_{1n} \text{ EXOR } a_{0(n-1)}) \text{ AND } (a_{0n} \text{ EXOR } b_{1(n-1)})] \end{aligned}$$

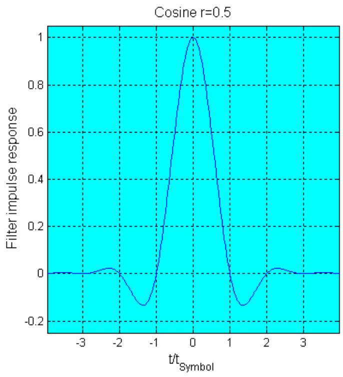
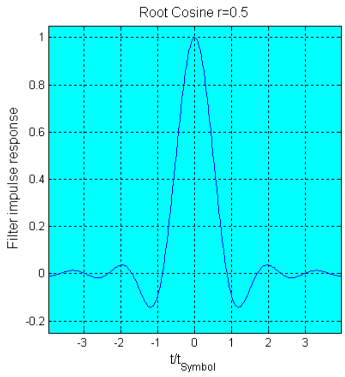
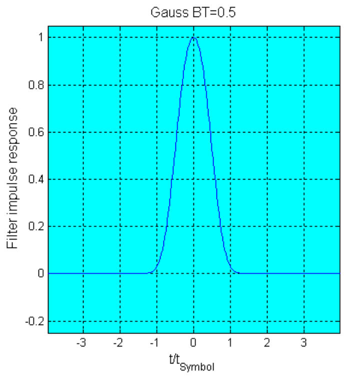
5.10.2.3 Baseband Filter - Custom Digital Mod

The R&S WinIQSIM2 offers a wide selection of predefined baseband filters. The filter characteristic for the selected filter is displayed in the "More..." submenu in the "Filter" menu section (see "Filter" on page 70). Additionally, a user defined filter can be selected (see "User filter" on page 60).

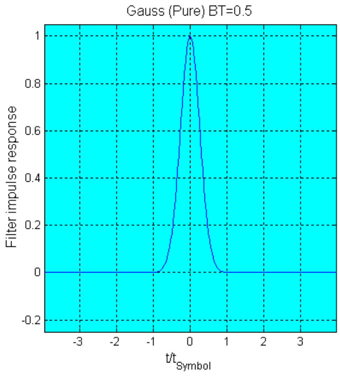
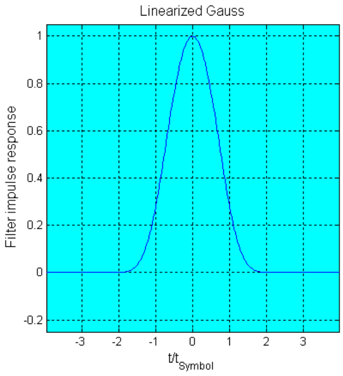
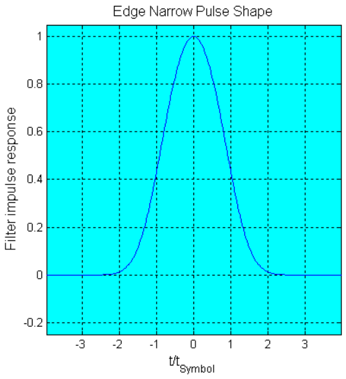
The following table shows the filters that are available, together with the associated parameters and IEC bus commands. The filter characteristic is displayed in graphical form.

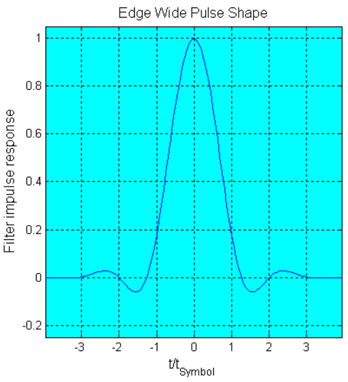
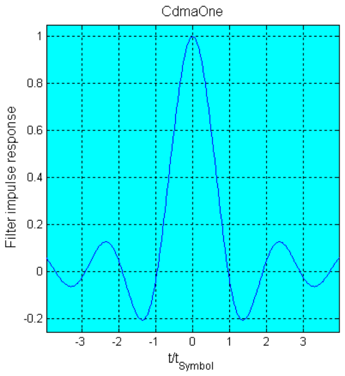
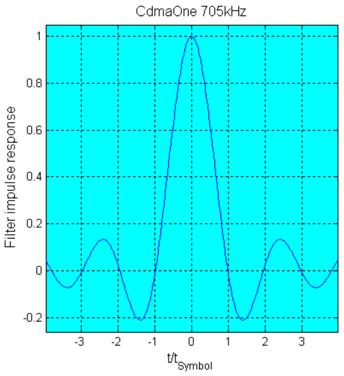
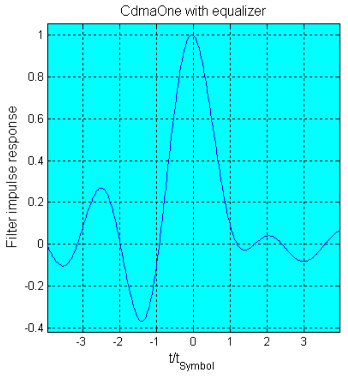
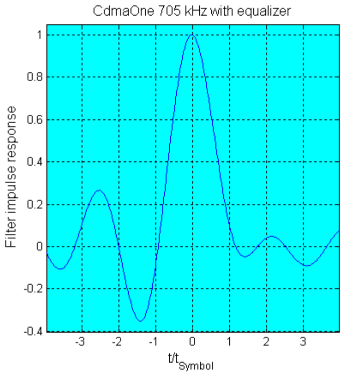
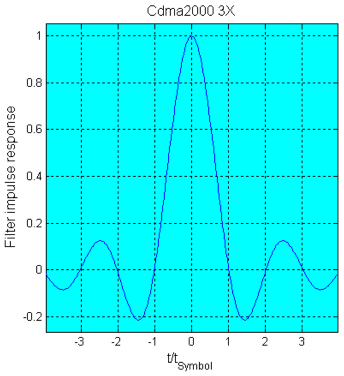
Table 5-9: Baseband filter

Cosine
Roll Off Factor
SOUR:BB:DM:FILT:TYPE COS
SOUR:BB:DM:FILT:PAR:COS 0.99
Bandwidth
SOUR:BB:DM:FILT:PAR:COS:BAND 7200
	Root Cosine
	Roll Off Factor
	`SOUR:BB:DM:FILT:TYPE RCOS`
	`SOUR:BB:DM:FILT:PAR:RCOS 0.99`
		Gauss (FSK)
		B x T
		`SOUR:BB:DM:FILT:TYPE GAUS`
		`SOUR:BB:DM:FILT:PAR:GAUS 2.5`

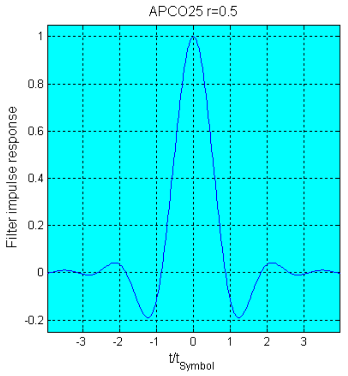
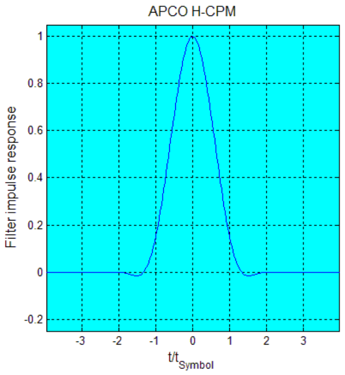
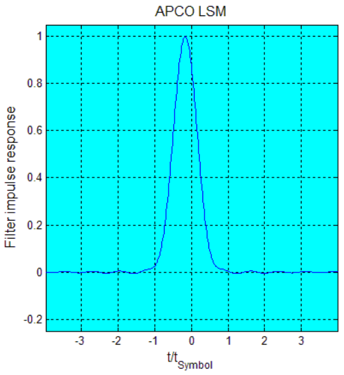
 	 	

Pure Gauss
B x T
`SOUR:BB:DM:FILT:TYPE PGA`
`SOUR:BB:DM:FILT:PAR:PGA 2.5`
	Gauss Linearized
	`SOUR:BB:DM:FILT:TYPE LGA`
		Edge Narrow Pulse Shape
		`SOUR:BB:DM:FILT:TYPE ENPShape`

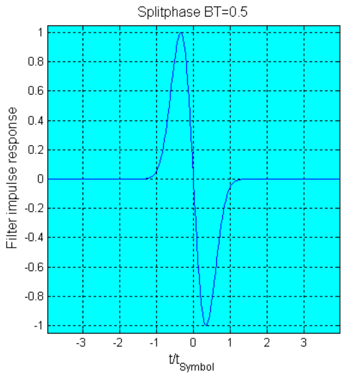
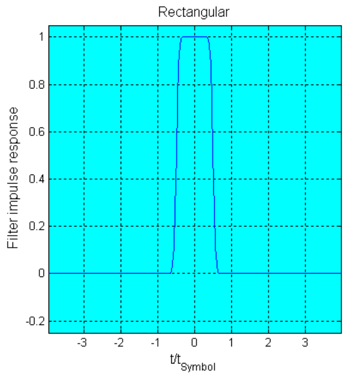
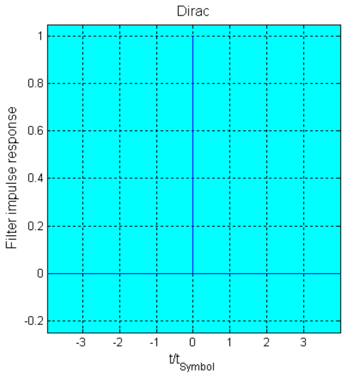
 	 	
Edge Wide Pulse Shape `SOUR:BB:DM:FILT:TYPE EWPShape`	cdmaOne `SOUR:BB:DM:FILT:TYPE CONE`	cdmaOne 705 kHz `SOUR:BB:DM:FILT:TYPE COF705`

		
cdmaOne + Equalizer `SOUR:BB:DM:FILT:TYPE COE`	cdmaOne 705 kHz + Equalizer `SOUR:BB:DM:FILT:TYPE COFE228`	cdma2000 3X `SOUR:BB:DM:FILT:TYPE DM3x`
		

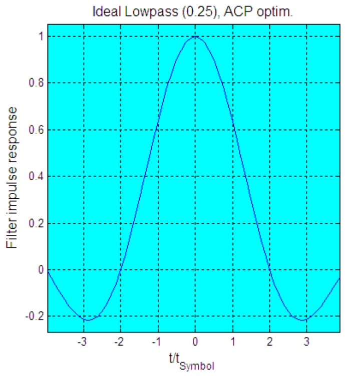
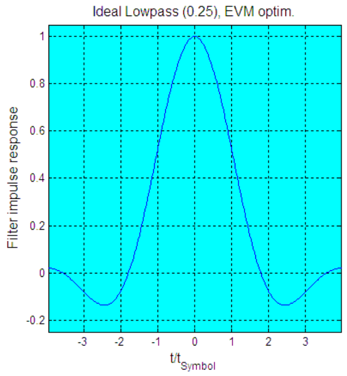
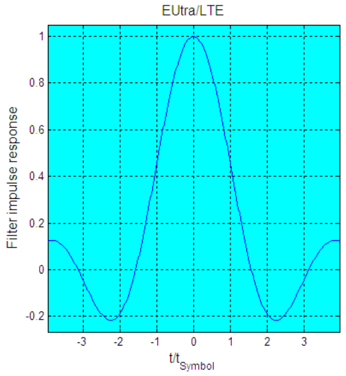
APCO25
Roll Off Factor
`SOUR:BB:DM:FILT:TYPE APCO25`
`SOUR:BB:DM:FILT:PAR:APC 0.5`
	APCO25 (H-CPM)
	`SOUR:BB:DM:FILT:TYPE APCO25Hcpm`
		APCO25 (LSM)
		`SOUR:BB:DM:FILT:TYPE APCO25Lsm`
		Gauss Cut Off Frequency
		SOUR:BB:DM:FILT:PAR:APCO25L:GAUS 25000
		Gauss Cut Off Frequency
		SOUR:BB:DM:FILT:PAR:APC025:LOWP 75000

Split Phase
BxT
`SOUR:BB:DM:FILT:TYPE  SPH`
`SOUR:BB:DM:FILT:PAR:SPH 0.15`
	Rectangular
	`SOUR:BB:DM:FILT:TYPE RECT`
		Dirac
		`SOUR:BB:DM:FILT:TYPE DIRac`

Lowpass ACP opt.
`SOUR:BB:DM:FILT:TYPE LPASs`
`SOUR:BB:DM:FILT:PAR:LAPS 0.5`
	Lowpass EVM opt.
	`BB:DM:FILT:TYPE LPASSEVM`
	`BB:DM:FILT:PAR:LAPSSEVM 0.5`
		EUTRA/LTE
		`SOUR:BB:DM:FILT:TYPE LTEFilter`

User filter

The user filter file must have extention `*.vaf` and can be created with the R&S filter wizard.

The filter wizard (filtwiz) is a tool from Rohde & Schwarz designed for creating filter files that can be imported on a R&S WinIQSIM2. Its main purpose is the conversion of user-defined finite impulse response (FIR) filters into the filter format (`*.vaf`). Beyond this filtwiz provides designs for standard filters, e.g. Root Raised Cosine, Gaussian.

The program was developed on a 32-bit Microsoft Windows platform under MATLAB.

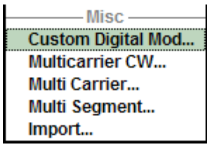
For more information, refer to the description "Introduction to "filtwiz" Filter Editor" on the Rohde&Schwarz Internet page.

5.10.2.4 Conflicting Settings - Custom Digital Mod

Having selected a modulation procedure, not every combination is possible when selecting the settings for the modulation parameters "Symbol Rate" and "Coding". These restrictions inevitably give rise to conflicting settings if a parameter is changed and leads to a prohibited combination.

A conflicting setting is indicated by a message on the Info line in the display. The R&S WinIQSIM2 displays the setting entered by the user, but the modulation signal actually generated does not correspond to this display. A conflict of settings can be triggered if the user changes a parameter inappropriately. The message disappears as soon as a conflict-free setting is entered.

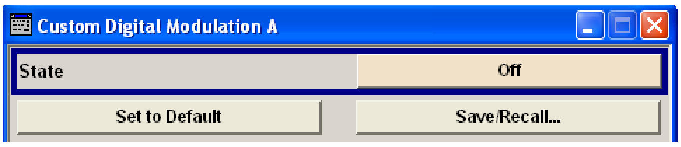
5.10.2.5 Custom Digital Mod User Interface



The menu for setting digital modulation can be opened either in the "Baseband" block or in the menu tree under "Baseband".

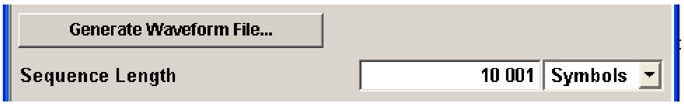
The "Custom Digital Modulation" menu enables direct selection of the data source, standard, symbol rate, coding, modulation type and filter. All other settings are entered in submenus which are called via "More" buttons.

The upper part of the menu is used for powering up digital modulation as well as for calling the default settings and user-defined standards.


Custom Digital Modulation A

State: Off

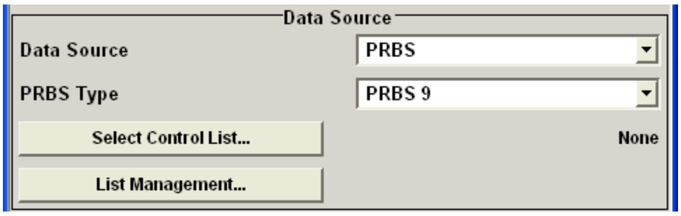
The "Generate Waveform File" opens the submenu for storing the current signal in a waveform file.



Sequence Length: 10 001 Symbols

The data source is selected and set in the "Data Source" section.

The "List Management" button opens a submenu for calling the data editor and the file manager.

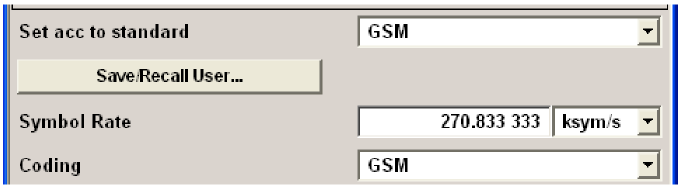

Data Source

Data Source: PRBS

PRBS Type: PRBS 9

Select Control List... None

The middle part of the menu is used for selecting the standard, the symbol rate and the coding.



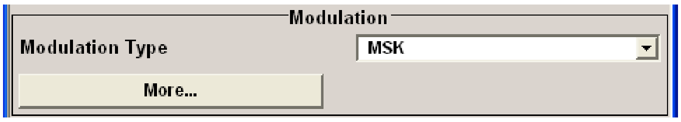
Set acc to standard GSM

Symbol Rate 270.833 333 ksym/s

Coding GSM

The modulation type is set in the "Modulation" section.

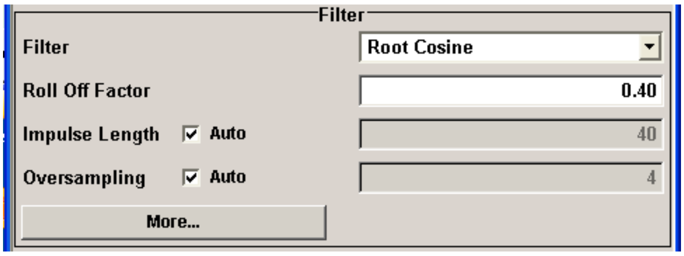
The "More" button opens a submenu that displays the mapping for the selected modulation.



Modulation

Modulation Type MSK

The filter is set in the "Filter" section. The "More" button opens a submenu where the filter characteristic of the selected filter is displayed.



Filter

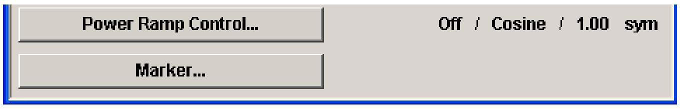
Filter Root Cosine

Roll Off Factor 0.40

Impulse Length ☑ Auto 40

Oversampling ☑ Auto 4

The buttons in the lower part of the menu open submenus for power ramping and for configuring markers.



Power Ramp Control... Off / Cosine / 1.00 sym

In each case the current setting is displayed next to the button.

General Settings

The upper part of the menu is used for powering up digital modulation as well as for calling the default settings and user-defined standards.

State - Digital Modulation

Enables/disables digital modulation.

Switching on digital modulation turns off all the other digital standards.

SCPI command:
`[:SOURce<hw>]:BB:DM:STATe` on page 232

Set To Default - Digital Modulation

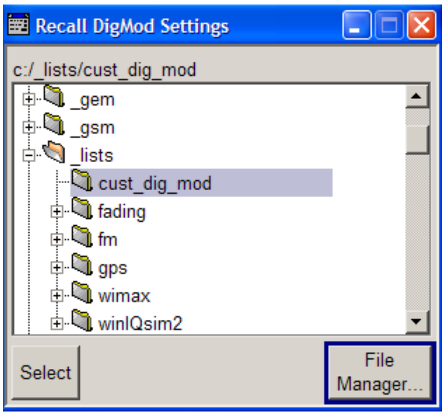
Calls default settings. The values are shown in the following table.

Parameter	**Value**
State	Not affected by Set to Default
Data Source	PRBS 9
Standard	GSM
Symbol Rate	270.833 ksymb/s
Coding	GSM
Modulation Type	MSK
Filter	Gauss (FSK)
Filter Parameter BxT	0.3
Power Ramp Control	
Attenuation	15 dB
Ramp Time	1 sym
Ramp Function	Cosine
Fall Delay	0
Rise Delay	0
Source	Internal
State	Off
Marker	Marker Mode On/Off Ratio
Clock	
Source	Internal
Mode	Symbol

SCPI command:
`[:SOURce<hw>]:BB:DM:PRESet` on page 230

Save/Recall - Digital Modulation
Calls the "Save/Recall" dialog.

From the "Save/Recall" dialog the "File Select" windows for saving and recalling the complete settings in the "Custom Digital Modulation" menu can be called.



The Digital Modulation Settings are stored as files with the predefined file extension `*.dm`. The file name and the directory they are stored in are user-definable.

The complete settings in the "Custom Digital Modulation" dialog are saved and recalled.

SCPI command:

`[:SOURce<hw>]:BB:DM:SETTing:CATalog` on page 253
`[:SOURce<hw>]:BB:DM:SETTing:DELete` on page 254
`[:SOURce<hw>]:BB:DM:SETTing:LOAD` on page 254
`[:SOURce<hw>]:BB:DM:SETTing:STORe` on page 254

Generate Waveform File - Digital Modulation

Opens the submenu for storing the current signal as ARB signal in a waveform file.

The file name is entered in the submenu. The file is stored with the predefined file extension `*.wv`. The file name and the directory it is stored in are user-definable.

SCPI command:

`[:SOURce<hw>]:BB:DM:WAVeform:CREate` on page 233

Sequence Length - Digital Modulation

Sets the sequence length of the signal in symbols. The signal is calculated in advance, saved as waveform file, and output in the selected arbitrary waveform generator.

Note: "Sequence Length" * "Oversampling" must not exceed the maximum number of samples in the arbitrary waveform generator.

SCPI command:

`[:SOURce<hw>]:BB:DM:SLENgth` on page 232

Fast Save

Determines whether the instrument performs an absolute or a differential storing of the settings.

Enable this function to accelerate the saving process by saving only the settings with values different to the default ones.

Note: This function is not affected by the "Preset" function.

SCPI command:

[:SOURce<hw>]:BB:DM:SETTing:STORe:FAST on page 254
[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:STORe:FAST on page 219
refer to the description of frimware options for the syntax of the corresponding SCPI commands

Data Source

The data source is selected and set in the "Data Source" section. The parameters offered depend on the data source selected. The More button opens a submenu for calling the data editor and the file manager.

Data Source - Digital Modulation

Selects the data source (see chapter 5.10.1, "Data and Signal Sources in Baseband", on page 48).

You may choose from the following data sources:

"All 0, All 1" 0 data or 1 data is internally generated.

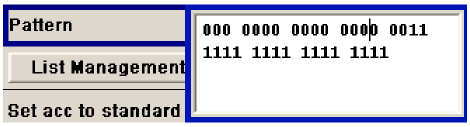
"PRBS, PRBS Type" PRBS data in accordance with the IUT-T with period lengths between 29-1 and 223-1 are internally generated.
The length is selected in the "PRBS Type" input box.

PRBS Type: PRBS 9

SCPI command:

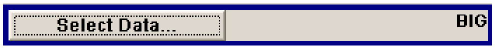
[:SOURce<hw>]:BB:DM:PRBS[:LENGth] on page 230

"Pattern" A user-definable bit pattern with a maximum length of 64 bits is internally generated.
The bit pattern is defined in the "Pattern" input box.



Pattern: 000 0000 0000 0000 0011 1111 1111 1111 1111

"Data List ..., Select Data" Data lists will be used.
Data lists can be generated internally in the data editor or externally.
Data lists are selected in the "File Select" window, which is called by means of the "Select Data" button.



The "File" "Manager" is used to transmit external data lists to the R&S Signal Generator, and can be called within every "File" Select window by means of the "File" "Manager" button.

SCPI command:

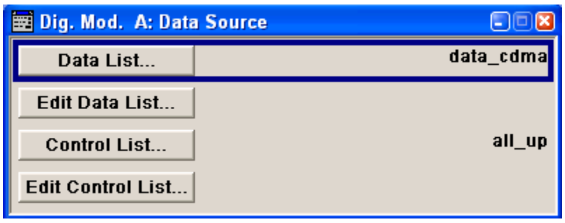
[:SOURce<hw>]:BB:DM:DLISt:SELect on page 251

SCPI command:

[:SOURce<hw>]:BB:DM:SOURce on page 230

List Management - Digital Modulation

Calls the menu for managing data and control lists (see "List Management" on page 72).



SCPI command:
n.a.

Standard, Symbol Rate, Coding

The middle part of the menu is used for selecting the standard, the symbol rate and the coding.

Set acc. to Standard - Digital Modulation

Selects a standard.

After selection, modulation parameters "Modulation Type", "Symbol Rate", "Filter" and "Coding" are automatically set in accordance with the standard.

If one of these parameters is subsequently altered, the display changes to User. The User setting can be saved to a file so that it can be recalled at some later time ("Save/Recall User.." button).

The following table shows the standards that are available, together with the associated settings of the modulation parameters.

Table 5-10: Standards - Custom Digital Modulation

Standard	Modulation	Symbol Rate	Filter	Coding	Parameter for SCPI command
APCO Phase1 C4FM	4FSK, Deviation 1.8 KHz	4.8 ksym/s	APCO25, roll off = 0.2	APCO25(FSK)	`APCOPH1C4fm`
APCO Phase1 CQPSK	pi/4-DQPSK	4.8 ksym/s	Cosine, roll off = 0.2, BW = 4.8 KHz	APCO25(PSK)	`APCOPH1CQpsk`
APCO Phase2 H-CPM	4FSK, Deviation 3 KHz	6 ksym/s	APCO25 (H-CPM)	APCO25(FSK)	`APCOPH2HCpm`
APCO Phase2 H-DQPSK	pi/4-DQPSK	6 ksym/s	Cosine, roll off = 1, BW = 7.2 KHz	APCO25(PSK)	`APCOPH2HDQpsk`
APCO Phase2 H-D8PSK Wide	pi/8-D8PSK	4 ksym/s	Cosine, roll off = 1, BW = 7.2 KHz	APCO25(8PSK)	`APCOPH2HD8PSKW`
APCO Phase2 H-D8PSK Narrow	pi/8-D8PSK	4 ksym/s	Cosine, roll off = 1, BW = 5 KHz	APCO25(8PSK)	`APCOPH2HD8PSKN`
APCO Phase1 LSM	pi/4-DQPSK	4.8 ksym/s	APCO25 (LSM), Gauss Cut Off = 2.04 KHz, Low-pass Cut Off = 6.2 KHz	APCO25(8FSK)	`APCOPH1Lsm`
APCO Phase1 WCQPSK	pi/4-DQPSK	4.8 ksym/s	Cosine, roll off = 1, BW = 7.2 KHz	APCO25(8FSK)	`APCOPH1Wcqpsk`

Standard	Modulation	Symbol Rate	Filter	Coding	Parameter for SCPI command
Bluetooth	2FSK, Deviation 160.0 kHz	1.0 Msym/s	Gauss, B*T = 0.5	OFF	BLUetooth
DECT	2FSK, Deviation 288.0 kHz	1.152 Msym/s	Gauss, B*T = 0.5	OFF	DECT
ETC (ARIB STD T55)	ASK, ASK Depth 100%	1.024 Msym/s	Split Phase, B*T = 2.0	OFF	ETC
GSM	MSK	270.833333 ksym/s	Gauss, B*T = 0.3	GSM	GSM
GSM EDGE	8PSK EDGE (3pi/8 8PSK)	270.833333 ksym/s	Gauss linear	OFF	GSMEdge
NADC	pi/4 DQPSK	24.3 ksym/s	Root Cosine, alpha = 0.35	NADC	NADC
PDC	pi/4 DQPSK	21.0 ksym/s	Root Cosine, alpha = 0.50	PDC	PDC
PHS	pi/4 DQPSK	192.0 ksym/s	Root Cosine, alpha = 0.50	PHS	PHS
TETRA	pi/4 DQPSK	18.0 ksym/s	Root Cosine, alpha = 0.35	TETRA	TETRa
WCDMA 3GPP	QPSK 45° Offset	3.84 Msym/s	Root Cosine, alpha = 0.22	WCDMA 3GPP	W3GPp
TD-SCDMA	QPSK 45° Offset	1.28 Msym/s	Root Cosine, alpha = 0.22	OFF	TCSCdma
cdma2000 Forward	QPSK	1.2288 Msym/s	cdmaOne + Equalizer	cdma2000	CFORward
cdma2000 Reverse	Offset QPSK	1.2288 Msym/s	cdmaOne	cdma2000	CREVerse
Worldspace	QPSK	1.84 Msym/s	Root Cosine, alpha = 0.40	OFF	WORLdspace
TFTS	pi/4 DQPSK	22.1 ksym/s	Root Cosine, alpha = 0.40	TFTS/ TETRA	TFTS

SCPI command:
`[:SOURce<hw>]:BB:DM:STANdard` on page 232

Save/Recall User - Digital Modulation

Calls the "Save/Recall User" menu.

From the "Save/Recall User Standard" menu the "File Select" windows for saving and recalling user-defined standards and the "File Manager" can be called.

User standards are stored as files with the predefined file extension `*.dm_stu`. The file name and the directory they are stored in are user-definable.

SCPI command:
`[:SOURce<hw>]:BB:DM:STANdard:ULISt:CATalog` on page 255
`[:SOURce<hw>]:BB:DM:STANdard:ULISt:DELete` on page 255
`[:SOURce<hw>]:BB:DM:STANdard:ULISt:LOAD` on page 255
`[:SOURce<hw>]:BB:DM:STANdard:ULISt:STORe` on page 256

Symbol Rate - Digital Modulation

Selects the symbol rate.

The value range is dependent on the selected modulation type.

When the modulation type changes, the range is automatically redefined. If the set symbol rate is outside this range, an error message is generated and the maximum value for the newly chosen modulation type is automatically set.

SCPI command:
`[:SOURce<hw>]:BB:DM:SRATe` on page 231

Coding - Digital Modulation

Selects the coding (see chapter 5.10.2.2, "Coding - Custom Digital Mod", on page 55).

The menu offers only the coding settings that are permissible for the chosen modulation type. The other coding methods are grayed out.

If the system is subsequently switched to a modulation type for which the selected coding is not available, coding is automatically set to OFF.

SCPI command:
`[:SOURce<hw>]:BB:DM:CODing` on page 240

Modulation

The modulation type is set in the "Modulation section". The parameters offered depend on what is currently selected. The "More..." button opens a submenu for displaying the user-defined mapping.

Modulation Type - Digital Modulation

Selects a modulation type.

The associated symbol mapping is displayed in the "More..." submenu (see chapter 5.10.2.1, "Modulation Types - Custom Digital Mod", on page 51).

If the "coding" that is set is not possible with the chosen modulation type, "coding" is set to "Off".

For selection "Variable FSK", the deviation of each symbol can be set in the "More..." submenu.

SCPI command:
`[:SOURce<hw>]:BB:DM:FORMat` on page 241

Load User Mapping - Digital Modulation

Selects user defined mapping table. This opens the "Select List File User Mapping" window in which the mapping table can be selected (see "User Mapping" on page 54). The button is only available if modulation type "USER" is selected.

The button "File Manager" calls the "File Manager" menu. The "File" "Manager" menu is used to display and delete files.

SCPI command:
`[:SOURce<hw>]:BB:DM:MLISt:SELect` on page 253
`[:SOURce<hw>]:BB:DM:MLISt:CATalog` on page 246
`[:SOURce<hw>]:BB:DM:MLISt:DELete` on page 252

ASK Depth - Digital Modulation

Sets the modulation depth m for ASK modulation.

m = (Amplitude$_{max}$-Amplitude$_{min}$) / (Amplitude$_{max}$+Amplitude$_{min}$)

SCPI command:

[:SOURce<hw>]:BB:DM:ASK:DEPTh on page 240

FSK Deviation - Digital Modulation

Sets the frequency deviation for FSK modulation. The range of values depends on the chosen symbol rate (see data sheet).

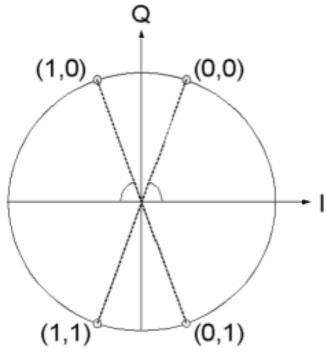
Whenever "MSK" is selected, the deviation corresponds to 1/4 of the symbol rate and cannot be set.

SCPI command:

[:SOURce<hw>]:BB:DM:FSK:DEViation on page 241

Angle Alpha

For AQPSK modulation, sets the angle alpha between the point (0,0) and the I axis.

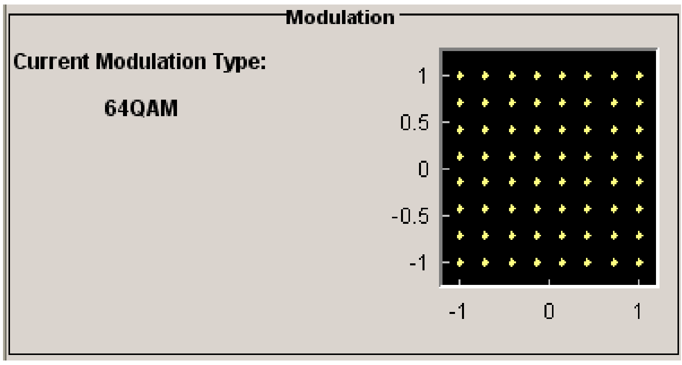


SCPI command:

[:SOURce<hw>]:BB:DM:AQPSk:ANGLe on page 239

More - Modulation Type - Digital Modulation

Calls the submenu which displays the mapping of the currently selected modulation type and the digital modulation delay in the case of an external data source.



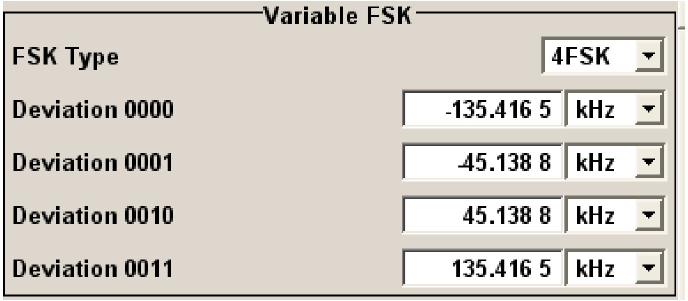
Modulation

Current Modulation Type:

64QAM

The "Modulation - CW Switching" section can be used to enable switching of the signal between modulated and unmodulated.

In case of selection "Variable FSK", the FSK type and the deviation for each symbols is set in the "Variable FSK" section.



Variable FSK

FSK Type	4FSK
Deviation 0000	-135.416 5 kHz
Deviation 0001	-45.138 8 kHz
Deviation 0010	45.138 8 kHz
Deviation 0011	135.416 5 kHz

FSK Type - Digital Modulation
(Variable FSK only)

Selects the FSK modulation type for selection "Variable FSK".

You may choose from 4FSK, 8FSK and 16FSK.

SCPI command:
`[:SOURce<hw>]:BB:DM:FSK:VARiable:TYPE` on page 242

Deviation xxxx - Digital Modulation
(Variable FSK only)

Set the deviation of the associated symbol. The number of symbols depends on the selected modulation type. The value of each symbol is indicated in binary format.

SCPI command:
`[:SOURce<hw>]:BB:DM:FSK:VARiable:SYMBol<ch>:DEViation` on page 241

Current Modulation Type - Digital Modulation
Displays the currently selected modulation type together with its associated mapping.

SCPI command:
n.a.

Filter

The filter is set in the Filter section.

Filter - Digital Modulation
Selects the baseband filter (see chapter 5.10.2.3, "Baseband Filter - Custom Digital Mod", on page 58).

SCPI command:
`[:SOURce<hw>]:BB:DM:FILTer:TYPE` on page 239

Filter Parameter - Digital Modulation
Sets the filter parameter.

The filter parameter offered ("Roll Off Factor" or "B x T") depends on the currently selected filter type.

SCPI command:

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25` on page 237

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:COSine[:ROLLoff]` on page 237

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:GAUSs` on page 237

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:PGAuss` on page 237

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:RCOSine` on page 237

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:SPHase` on page 237

Cut Off Frequency Factor - Digital Modulation

(available for filter parameter Lowpass and APCO25 (LSM) only)

Sets the value for the cut off frequency factor. The cut off frequency of the filter can be adjusted to reach spectrum mask requirements.

SCPI command:

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:LPASs` on page 237

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:LPASSEVM` on page 237

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25Lsm:GAUSs` on page 239

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25Lsm:LOWPass` on page 239

Bandwidth - Digital Modulation

Determines the bandwidth of the cosine filter, so that the function H(f) = 0 is fulfilled for *f >= (1+RollOff)*Bandwidth/2*.

SCPI command:

`[:SOURce<hw>]:BB:DM:FILTer:PARameter:COSine:BANDwidth` on page 238

Impulse Length - Digital Modulation

Displays the number of filter tabs. If the check box is activated, the most sensible parameter values are selected. The value depends on the coherence check. If the check box is deactivated, the values can be changed manually.

SCPI command:

`[:SOURce<hw>]:BB:DM:FILTer:ILENgth:AUTO` on page 236

`[:SOURce<hw>]:BB:DM:FILTer:ILENgth` on page 236

Oversampling - Digital Modulation

Determines the upsampling factor. If the check box is activated, the most sensible parameter values are selected. The value depends on the coherence check. If the check box is deactivated, the values can be changed manually.

SCPI command:

`[:SOURce<hw>]:BB:DM:FILTer:OSAMpling:AUTO` on page 237

`[:SOURce<hw>]:BB:DM:FILTer:OSAMpling` on page 237

More - Filter - Digital Modulation

Calls the menu for displaying the filter characteristic of the currently selected filter.

Load User Filter- Filter - Digital Modulation

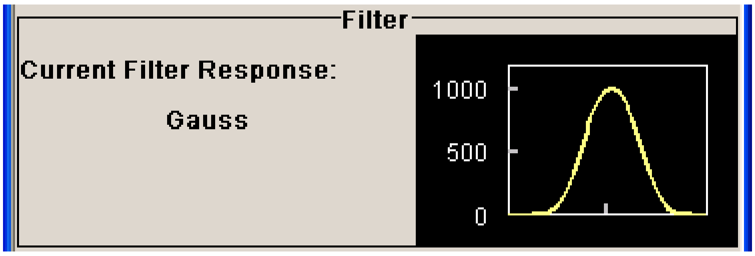
Calls the menu "Select List File User Filter" for selecting a user-defined filter file with extension *.vaf (see "User filter" on page 60). The button is only available if filter type "USER" is selected.

The button "File" "Manager" calls the "File" "Manager" menu. The "File" "Manager" menu is used to display and delete files.

SCPI command:
[:SOURce<hw>]:BB:DM:FLISt:SELect on page 252
[:SOURce<hw>]:BB:DM:FLISt:CATalog on page 246
[:SOURce<hw>]:BB:DM:FLISt:DELete on page 251

Current Filter Response

Displays the filter characteristic of the currently selected filter.



SCPI command:
n.a.

Power Ramp

The lower part of the Custom Digital Modulation menu is used for setting markers and for power ramp settings.

Power Ramp Control - Digital Modulation

Calls the power ramp control menu (see "Power Ramp Control" on page 79).

SCPI command:
n.a.

Marker - Digital Modulation

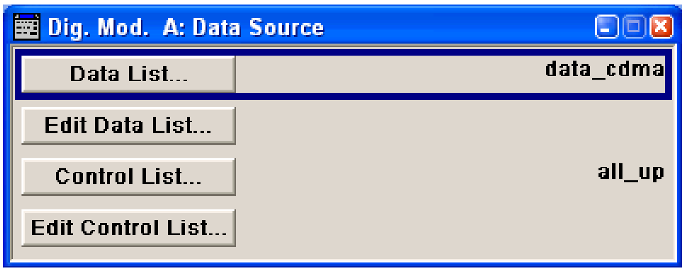
Calls the "Marker" menu, used to configure the marker output signals.

See "Marker Mode" on page 82.

SCPI command:
n.a.

List Management

The "List Management" menu is called from the "Custom Digital Modulation" main menu.


Dig. Mod. A: Data Source

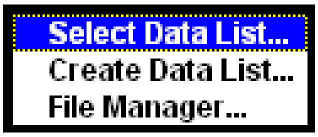
Data List... data_cdma

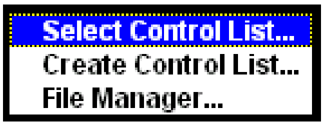
Edit Data List...

Control List... all_up

Edit Control List...

The data and control lists are selected and created in the "File Select" menu which is called up by means of the "Data List..." and "Control List..." buttons.





The "File Manager" is used to copy, rename and delete files and to create directories.

To ensure that the selected data or control list is used for generating the digital signal, the list must be selected as the data source:

Data	Parameter	Selection
Digital data	"Source"	DList
Marker	"Marker Mode"	CList
Control signals CW, Level Attenuation and Burst Gate	"Source"	Internal

The data editor is called using the "Edit Data List..." and "Edit Control List ..." buttons. The contents of the selected list are displayed.

Select Data/Control List - Digital Modulation

Selects data/control list. This opens the "File Select" window in which the data/control list can be selected.

SCPI command:

`[:SOURce<hw>]:BB:DM:DLISt:SELect` on page 251

`[:SOURce<hw>]:BB:DM:CLISt:SELect` on page 249

Create Data/Control List - Digital Modulation

Creates new data/control list. This opens the "File Select" window in which the data/control list can be created.

The file name has to be entered in field "File Name:". The new list contains no data, it can be edited in the list editor.

SCPI command:

`[:SOURce<hw>]:BB:DM:DLISt:SELect` on page 251

`[:SOURce<hw>]:BB:DM:CLISt:SELect` on page 249

File Manager - Digital Modulation

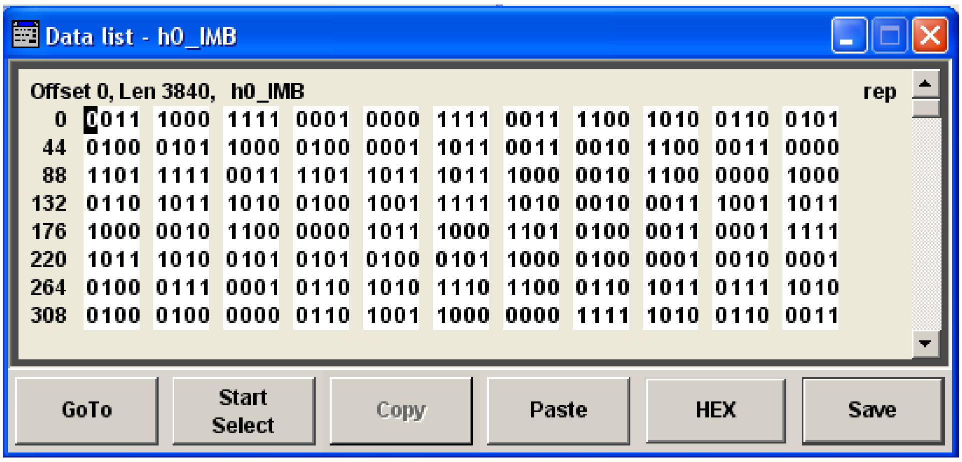
Calls the "File Manager".

The "File Manager" is used to copy, delete and rename files and to create new directories.

SCPI command:
`[:SOURce<hw>]:BB:DM:DLISt:SELect` on page 251
`[:SOURce<hw>]:BB:DM:DLISt:COPY` on page 246
`[:SOURce<hw>]:BB:DM:DLISt:DELete` on page 250
`[:SOURce<hw>]:BB:DM:CLISt:SELect` on page 249
`[:SOURce<hw>]:BB:DM:CLISt:COPY` on page 246
`[:SOURce<hw>]:BB:DM:CLISt:DELete` on page 248

Data List Editor

The "Data List Editor" for editing the selected data list is called up in the "List Management" submenu of the "Custom Digital Modulation" main menu by means of the "Edit Data List..." button.



```
Offset 0, Len 3840,  h0_IMB                                             rep
  0  0011 1000 1111 0001 0000 1111 0011 1100 1010 0110 0101
 44  0100 0101 1000 0100 0001 1011 0011 0010 1100 0011 0000
 88  1101 1111 0011 1101 1011 1011 1000 0010 1100 0000 1000
132  0110 1011 1010 0100 1001 1111 1010 0010 0011 1001 1011
176  1000 0010 1100 0000 1011 1000 1101 0100 0011 0001 1111
220  1011 1010 0101 0101 0100 0101 1000 0100 0001 0010 0001
264  0100 0111 0001 0110 1010 1110 1100 0110 1011 0111 1010
308  0100 0100 0000 0110 1001 1000 0000 1111 1010 0110 0011
```

Data List Editor

A list of binary values with a maximum length of 2^31 bits can be entered in the "Data List Editor". This value corresponds to a file size of approx. 268 Mbyte.

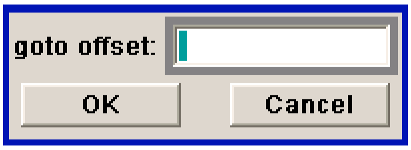
To increase readability, the bits are displayed in groups of four. The current cursor position, the length of the list and the list file name are displayed above the list. The offset starts with the value 0 which corresponds to the bit position on the left side of the first row, i.e. the beginning of the list. On the left edge of the editor, the last three offset positions are specified at the beginning of the row.

An existing list can be edited in the insert or overwrite mode.

SCPI command:
`[:SOURce<hw>]:BB:DM:DLISt:SELect` on page 251
`[:SOURce<hw>]:BB:DM:DLISt:DATA` on page 249
`[:SOURce<hw>]:BB:DM:DLISt:DATA:APPend` on page 250

GoTo ← Data List Editor

Opens the entry window for the bit position. The cursor marks the bit at the selected position.



SCPI command:
n.a.

Start Select ← Data List Editor
Defines the current cursor position as the start position for the range to be marked. The stop position is defined by entering an offset under "GoTo".

When a start position has been activated, the button will be re-labelled to "Undo Select". When the button is clicked now, the selected range will be deactivated.

SCPI command:
n.a.

Copy ← Data List Editor
Copies the selected values.

SCPI command:
n.a.

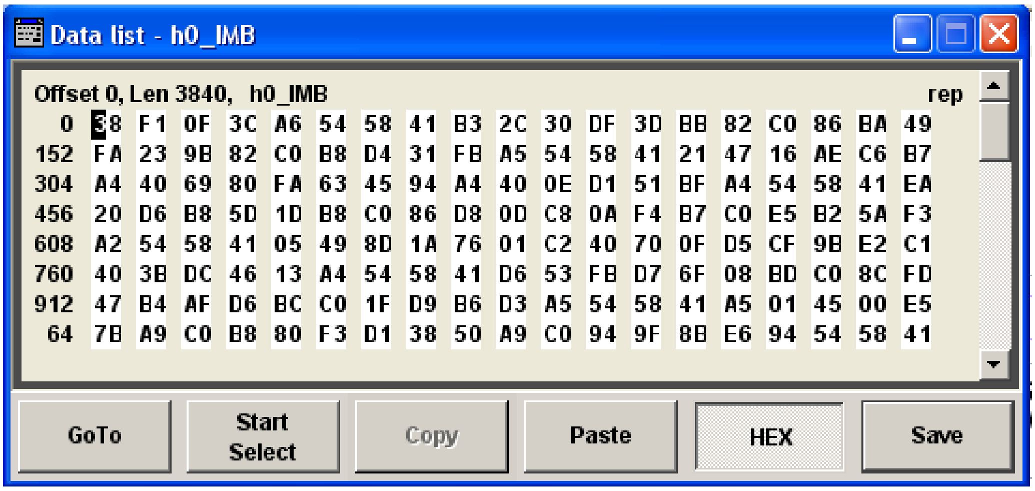
Cut ← Data List Editor
Cuts the selected values.

SCPI command:
n.a.

Paste ← Data List Editor
Pastes the values that have been copied or cut before.

SCPI command:
n.a.

Hex ← Data List Editor
Switchover to hexadecimal display.



```
Offset 0, Len 3840,   h0_IMB                                     rep
  0  38 F1 0F 3C A6 54 58 41 B3 2C 30 DF 3D BB 82 C0 86 BA 49
152  FA 23 9B 82 C0 B8 D4 31 FB A5 54 58 41 21 47 16 AE C6 B7
304  A4 40 69 80 FA 63 45 94 A4 40 0E D1 51 BF A4 54 58 41 EA
456  20 D6 B8 5D 1D B8 C0 86 D8 0D C8 0A F4 B7 C0 E5 B2 5A F3
608  A2 54 58 41 05 49 8D 1A 76 01 C2 40 70 0F D5 CF 9B E2 C1
760  40 3B DC 46 13 A4 54 58 41 D6 53 FB D7 6F 08 BD C0 8C FD
912  47 B4 AF D6 BC C0 1F D9 B6 D3 A5 54 58 41 A5 01 45 00 E5
 64  7B A9 C0 B8 80 F3 D1 38 50 A9 C0 94 9F 8B E6 94 54 58 41
```

Each four bits are displayed as a hexadecimal value: To increase readability, the hexadecimal values in turn are displayed in pairs of two. The hex functions are automatically assigned to the numeric keys at the front panel.

SCPI command:
n.a.

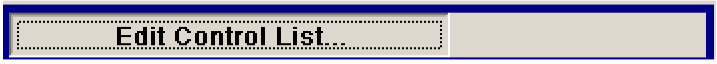
Save (Data List) ← Data List Editor

Saves the changes made to the Data List file selected for editing.

SCPI command:
n.a.

Control and Marker List Editor

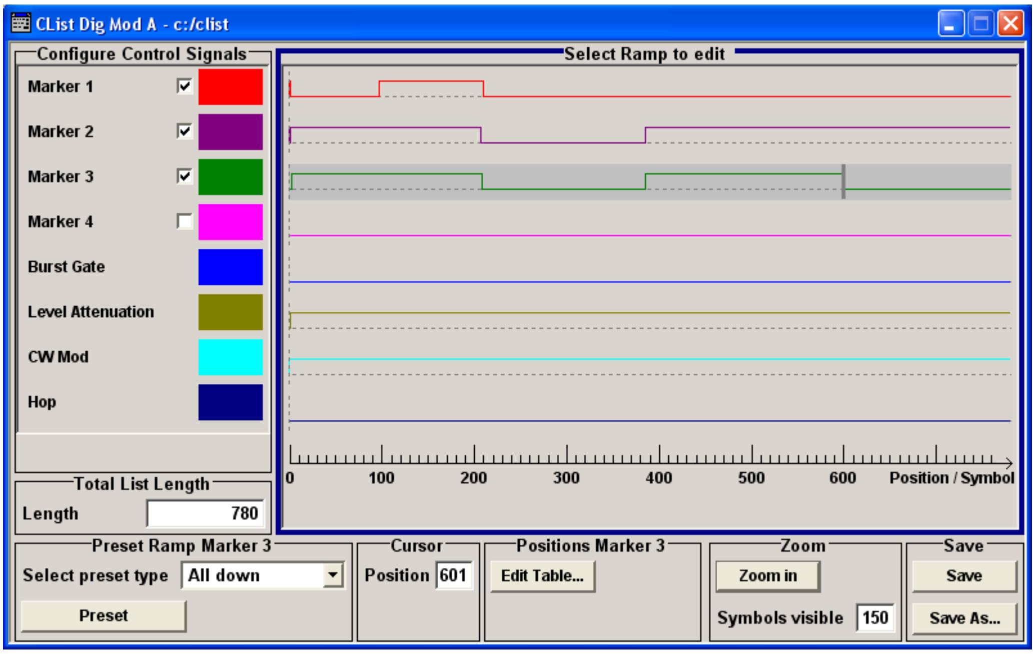
The "Control and Marker List Editor" for editing the selected control list is called up in the "List Management" submenu of the "Custom Digital Modulation" main menu by means of the "Edit Control List"... button.



The available marker signals and the CW, Hop, Burst Gate and Level Attenuation control signals can be defined in the "Control and Marker List Editor".

The marker signals defined in the Control and Marker List Editor are activated by selecting the Marker mode "CList" in the "Marker" menu.

The available marker/control signals are color-coded. In the left "Configure Control Signal" section, each individual signal is assigned a colour; a check in the check box shows the marker for which the "CList" marker type has been selected and the control signal for which the "Internal" source has been selected.



Length 780

Select preset type All down

Position 601

Symbols visible 150

In the "Select Ramp to Edit" section the signal characteristics are graphically displayed. The scaling of the x-axis is always adapted to the overall length of the control list to provide constant overview of all defined ramps.

The ramps can be assigned the exact bit position in the signal by means of

- The bit scale below the marker/control signal characteristic.
- The display of the current cursor position in the "Cursor" menu section if the cursor marks the ramp.

The ramps can be set either graphically in the "Select Ramp to Edit" section or in the table of the "Positions <Signal>" section in the lower right corner. To make the setting easy, a selection of preset ramp characteristics is offered in the "Preset Ramp <Signal>" section.

Edit Control List
The configuration of the currently selected control list is displayed. This list is either selected in the "File Select" menu ("List Management "submenu by means of the "Control List... "button) or via the remote-control command.

SCPI command:
`[:SOURce<hw>]:BB:DM:CLISt:SELect` on page 249
`[:SOURce<hw>]:BB:DM:CLISt:DATA` on page 247

Configure Control Signal - Digital Modulation
Displays the colour the marker/control signal has been assigned.

Displays whether the "CList" marker type has been selected in the Trigger/Marker/Clock menu for this marker.

Burst Gate and Level Attenuation are set in the "Power Ramping" submenu, CW in the "Modulation" submenu.

The source "CList/Internal" for the individual marker/control signal can be selected here as well and will then be used in the associated menus.

SCPI command:
n.a.

Select Ramp to Edit - Digital Modulation
Graphically edit marker/control signals.

For this purpose, the cursor is set to the position where a ramp is required. The ramp is generated by pressing Enter (e.g. clicking on the rotary knob). Any number of ramps can be defined per marker. Each of the generated ramp positions will be saved even if the definition of another ramp produces a low/low or high/high transition. These ramps are displayed as dashed lines.

Existing ramps can be shifted after the cursor has been placed on the ramp and Enter has been pressed - it then changes colour twice. The ramp is shifted by using the cursor keys or the rotary knob. The new position is determined by pressing Enter again.

Ramps can be deleted by means of the BACKSPACE key after the cursor has been placed on the ramp.

SCPI command:
n.a.

Total List Length - Digital Modulation
Enters the length of the definition range of the control list in bits. The starting value is always bit 0. The entire definition range is displayed, i.e. the bit scale is adapted to the entry.

With very long control lists, the displayed area can be zoomed around the current cursor position ("Zoom in" button).

The preset functions set the ramp in the center of the currently selected area.

If the definition range is decreased, the ramps outside the range are lost.

When used, the control list is always repeated over the length of the definition range if the length of the data list exceeds the length of the control list.

SCPI command:
n.a.

Preset Type - Digital Modulation
Activates presetting for the ramp characteristic of the selected control signal. The presetting is selected with "Preset Type" and activated by means of the "Preset" button.

You can select from:

"All Up "	The marker/control signal is continuously high.
"All Down"	The marker/control signal is continuously low.
"Ramp Up"	The marker/control signal contains a ramp from low to high. The ramp is shifted to the center of the displayed signal area and can subsequently be shifted as required.
"Ramp Down"	The marker/control signal contains a ramp from high to low. The ramp is shifted to the center of the displayed signal area and can subsequently be shifted as required.
"Ramp Up/ Down"	The marker/control signal contains a ramp from low to high and from high to low. The ramps are symmetrically shifted around the center of the displayed signal area and can subsequently be shifted as required.
"Ramp Down/ Up"	The marker/control signal contains a ramp from high to low and from low to high. The ramps are symmetrically shifted around the center of the displayed signal area and can subsequently be shifted as required.

SCPI command:
n.a.

Cursor Position - Digital Modulation
Enters the cursor position.

In the graphic display, the cursor is positioned according to the entry.

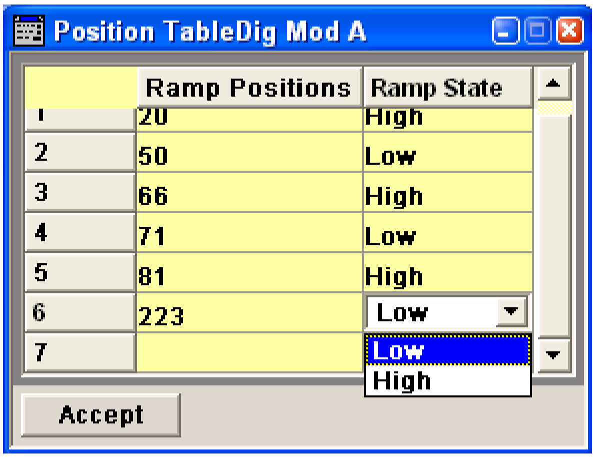
Vice versa, graphically shifting the cursor will change the displayed value.

If the entered value exceeds the selected length of the definition range, the length is adjusted automatically.

SCPI command:
n.a.

Positions Control Signal - Digital Modulation
Opens table by using the "Edit Table .. "button.

The ramps of the selected signal can be edited in the table. When the table is opened, the current configuration of the selected marker/control signal is displayed.



	Ramp Positions	Ramp State
1	20	High
2	50	Low
3	66	High
4	71	Low
5	81	High
6	223	Low
7		

The bit position is specified in the "Ramp Position" column, the high or low signal status in the "Ramp State" column. At the end of the list, there is always a blank row for entering new values.

The changes are accepted in the graphic display after pressing the "Accept" button.

SCPI command:
n.a.

Zoom/Visible - Digital Modulation

Zooms the displayed area of the control list. The designation of the button changes from "Zoom in" to "Zoom out".

With long control lists, the displayed area can be zoomed around the current "Cursor Position".

Parameter "Visible/Bits Visible" determines the number of symbols/bits to be displayed.

Ramps outside the displayed area are not lost by zooming.

SCPI command:
n.a.

Save/Save As - Digital Modulation

Pressing the "Save" button saves the changes made to the CList file selected for editing.

Selecting the "Save As" button creates a new control list. This opens the "Create Lists Files Which Name?" window in which the control list can be created.

The file name has to be entered in field "File Name."

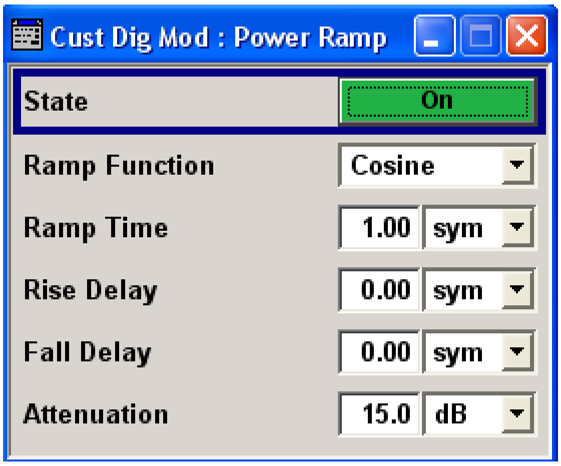
SCPI command:
n.a.

Power Ramp Control

The Power Ramp Control menu is accessed via the "Custom Digital Modulation" main menu.

The menu is used to set the power ramping. Control signals "Burst" and "Lev_Att" are used to control power ramping (see "Power Ramping and Level Attenuation" on page 50).

Power ramping is possible up to a symbol rate of 5 MHz. If a higher symbol rate is set, power ramping is automatically switched off and an error message is output.



State	On
Ramp Function	Cosine
Ramp Time	1.00 sym
Rise Delay	0.00 sym
Fall Delay	0.00 sym
Attenuation	15.0 dB

State - Power Ramp Control - Digital Modulation
Enables/disables power ramping.

SCPI command:
`[:SOURce<hw>]:BB:DM:PRAMp[:STATe]` on page 244

Ramp Function - Digital Modulation
Enters the form of the transmitted power, i.e. the shape of the rising and falling edges during power ramp control.

"Linear" The transmitted power rises and falls linear fashion.

"Cosine" The transmitted power rises and falls with a cosine-shaped edge. This gives rise to a more favorable spectrum than the "Linear" setting.

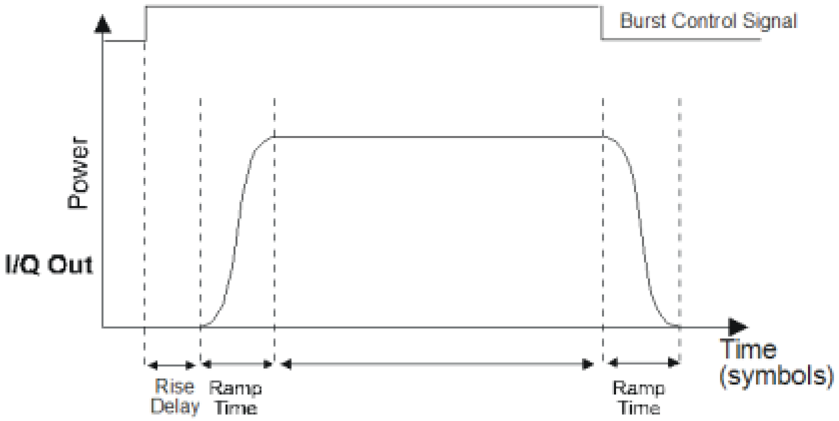
SCPI command:
`[:SOURce<hw>]:BB:DM:PRAMp:SHAPe` on page 243

Ramp Time - Digital Modulation
Enters the power ramping rise time and fall time for a burst. The setting is expressed in symbols.

SCPI command:
`[:SOURce<hw>]:BB:DM:PRAMp:TIME` on page 243

Rise Delay - Power Ramp Control - Digital Modulation
Sets the offset in the rising edge of the envelope at the start of a burst.

A positive value gives rise to a delay (the envelope length decreases) and a negative value causes an advance (the envelope length increases). The setting is expressed in symbols.

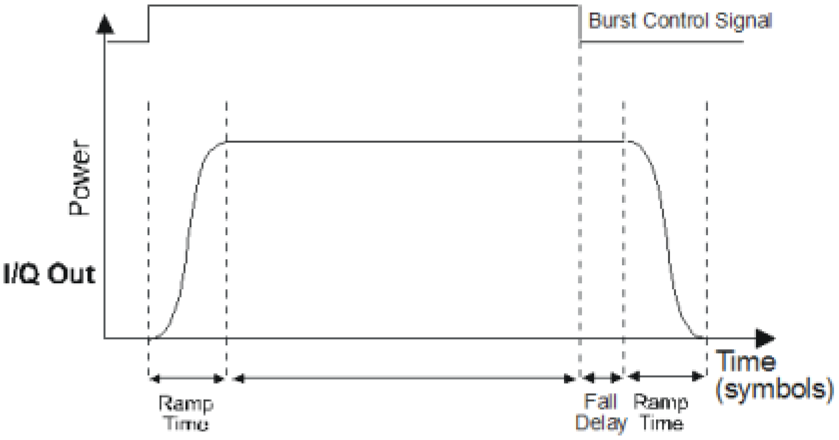


SCPI command:
[:SOURce<hw>]:BB:DM:PRAMp:RDELay on page 243

Fall Delay - Power Ramp Control - Digital Modulation

Sets the offset in the falling edge of the envelope at the end of a burst.

A positive value gives rise to a delay, the envelope length increases and a negative value causes an advance (the envelope length decreases). The setting is expressed in symbols.



SCPI command:
[:SOURce<hw>]:BB:DM:PRAMp:FDELay on page 243

Attenuation - Power Ramp Control - Digital Modulation

Sets the level attenuation relative to the average level for the signal ranges in which the level is set to "attenuated" (LEV_ATT control signal).

SCPI command:
[:SOURce<hw>]:BB:DM:PRAMp:ATTenuation on page 242

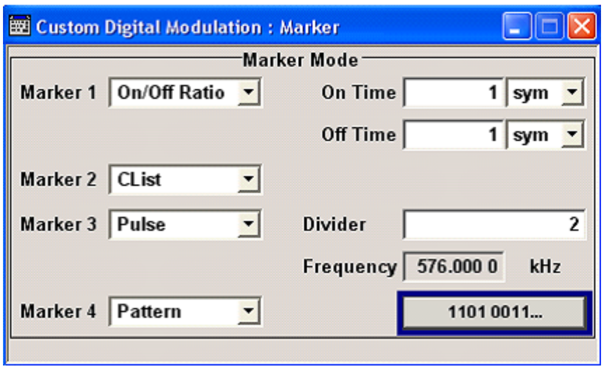
Trigger/Marker/Clock

The "Trigger/Marker/Clock" dialogis accessed via the "Custom Digital Modulation" main menu.

The "Marker Mode" section is where the marker signals at the MARKER output connectors are configured.

Marker Mode

The marker output signal for synchronizing external instruments is configured in the Marker Settings section "Marker Mode".



Custom Digital Modulation : Marker

Marker Mode

Marker 1: On/Off Ratio — On Time: 1 sym; Off Time: 1 sym

Marker 2: CList

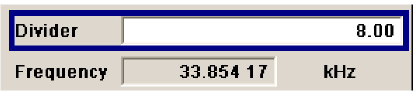
Marker 3: Pulse — Divider: 2; Frequency: 576.000 0 kHz

Marker 4: Pattern — 1101 0011...

Marker x - Digital Modulation
Selects a marker signal for the associated MARKER output.

"CList" A marker signal that is defined in the selected control list is generated.

"Pulse" A regular marker signal is generated. The clock frequency is defined by entering a divider. The frequency is derived by dividing the sample rate by the divider. The input box for the divider opens when "Pulse "is selected, and the resulting pulse frequency is displayed below it.



Divider: 8.00

Frequency: 33.854 17 kHz

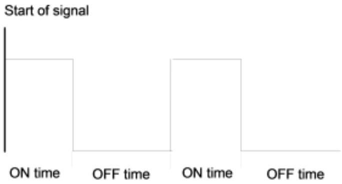
SCPI command:
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PULSe:DIVider on page 235

"Pattern" A marker signal that is defined by a bit "pattern" is generated. The "pattern" has a maximum length of 64 bits and is defined in an input field which opens when "pattern" is selected.

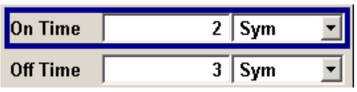
0000 0000

SCPI command:
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PATTern on page 235

"ON/OFF ratio" A regular marker signal that is defined by an ON/OFF ratio is generated. A period lasts one ON and OFF cycle.



The ON time and OFF time are each expressed as a number of symbols and are set in an input field which opens when "ON/OFF ratio" is selected.



On Time	2	Sym
Off Time	3	Sym

SCPI command:
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:OFFTime on page 234
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:ONTime on page 234

SCPI command:
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:MODE on page 233

5.10.3 Multi Carrier Continuous Wave

The R&S WinIQSIM2 can generate a Multi Carrier CW signal with user-definable offset from carrier, based on a selection of up to 8192 unmodulated carriers. Each carrier can be separately set and switched on, or multiple carriers can be jointly configured. Automatic start phase setting is provided in order to minimize the crest factor.

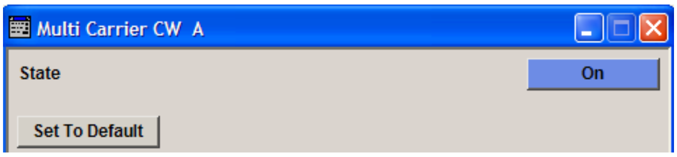
Multi Carrier CW signals can be very easily configured as broadband test signals and used for such purposes as receiver tests.

To access the dialog for setting a Multi Carrier CW signal, select "Baseband > Multi Carrier CW" or use MENU key under "Baseband".

5.10.3.1 Multi Carrier CW User Interface

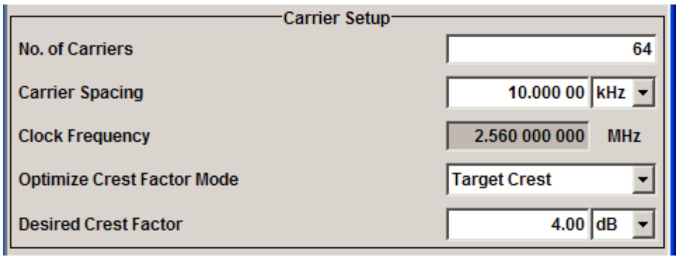
The "Multi Carrier CW" dialog is divided into the following sections.

The upper part of the menu is used for powering up the Multi Carrier CW and calling the default settings.



State: On

The "Carrier Setup" section is used to configure the Multi Carrier CW signal.



Carrier Setup

No. of Carriers	64
Carrier Spacing	10.000 00 kHz
Clock Frequency	2.560 000 000 MHz
Optimize Crest Factor Mode	Target Crest
Desired Crest Factor	4.00 dB

The "Table Setup Assistant" section can be used to set a selectable carrier range.

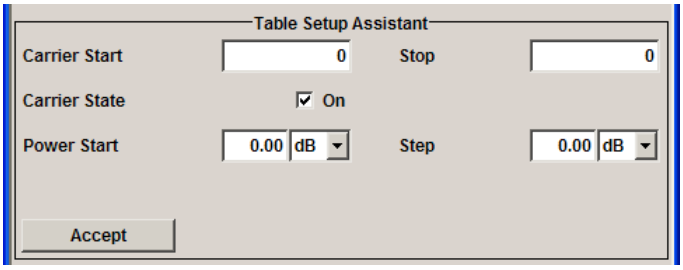
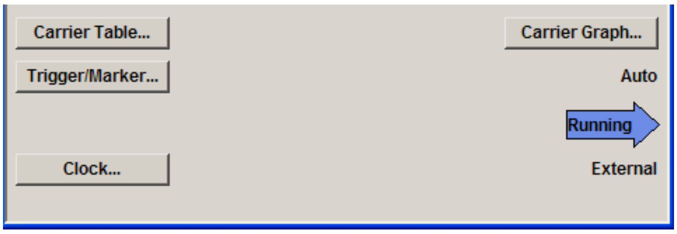


Table Setup Assistant

Carrier Start	0	Stop	0
Carrier State	☑ On		
Power Start	0.00 dB	Step	0.00 dB

The buttons in the lower part of the menu open submenus for defining carrier tables and for configuring markers.

The carrier settings can be visualized in the graphical "Carrier Graph" dialog.



General Settings for Multi Carrier CW

The upper part of the menu is used for powering up the Multi Carrier CW signal and calling the default settings.

State
Enables/disables the Multi Carrier CW.

Switching on this standard turns off all the other digital standards and digital modulation modes.

Since Multi Carrier CW signals are computed in arbitrary waveform mode, changes to the settings of individual carriers are not adopted until the Accept button is pressed. This applies to the settings in the "Table Setup Assistant" section and the "Carrier Table" dialog.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:STATe` on page 267

Set to Default
Calls default settings. The values are shown in the following table.

Parameter	Value
State	Not affected by "Set to Default"
Carrier Setup	
Number of Carriers	64
Carrier Spacing	10 kHz
Optimize Crest Factor	Chirp
Desired Crest Factor	3 dB
Trigger	
Mode	Auto
Source	Internal
Ext. Delay	0
Ext. Inhibit	0
Marker	
Channel 1...4	Restart
Clock	
Source	Internal
Multi Channel Setup	
Start Carrier	0
Stop Carrier	0
State	ON
Power	0 dB
Power Step	0 dB
Initial Phase	0°
Phase Step	0°
Channel Setup	
State	ON
Phase	0°
Power	0 dB

SCPI command:
`[:SOURce<hw>]:BB:MCCW:PRESet` on page 267

Carrier Setup

The "Carrier Setup" section is used to configure the Multi Carrier CW.

Number of Carriers

Sets the number of carriers for the Multi Carrier CW signal.

By default the multi carrier table already lists 64 carriers that are preset to the settings State = ON, Power = 0 dB, Phase = 0°.

When entering fewer carriers than the table contains, the approach is generally to delete the superfluous entries from the table, and when entering more carriers than the table contains the missing entries are usually added at the end of the table.

SCPI command:

`[:SOURce<hw>]:BB:MCCW:CARRier:COUNt` on page 267

Carrier Spacing

Sets the spacing between carriers for the Multi Carrier CW signal.

The carriers are arranged symmetrically around the HF carrier.

The total bandwidth is calculated as follow:

Total Bandwidth = ("Number of Carriers" - 1) * "Carrier Spacing"

The result must not exceed the system bandwidth of the instrument (see data sheet).

SCPI command:

`[:SOURce<hw>]:BB:MCCW:CARRier:SPACing` on page 270

Clock Frequency

Displays the clock rate at which the multi carrier signal is output by the arbitrary waveform generator. The output clock rate depends on the number of carriers and the selected carrier offset.

The value indicates the resolution during the marker generation.

SCPI command:

`[:SOURce<hw>]:BB:MCCW:CLOCk` on page 272

Optimize Crest Factor Mode

Selects the mode for automatically minimizing the crest factor.

The carrier start phases are automatically set to this.

The crest factor represents the ratio of the peak voltage value to the rms voltage value. The higher the crest factor and resulting dynamics of a signal, the greater the requirement for a power amplifier fed by the signal to be linear.

A very high crest factor arises when the carriers have an identical start phase, since the carriers are periodically superposed and very high peak voltages occur in relation to the rms voltage values.

Methods of reducing the crest factor differ with regard to both the optimization achievable and the time required for computation.

"Off" There are no automatic settings for minimizing the crest factor. The "Phase" setting has an effect.

"Chirp" Very rapid crest factor optimization regardless of the number of carriers. A minimal crest factor of < 3 dB is only obtained for multi carrier signals in which all carriers are switched on and the power of the carriers is identical. In a configuration which differs from this, the achievable crest factor is worse.

"Target Crest" Optimization of the crest factor to a desired value for all carrier configurations. The optimization time depends on the number of carriers and the desired crest factor. Computation time increases only when the number of carriers exceeds 256 and the crest factor is below 4 dB. The desired value can be entered in "Desired Crest Factor".

Note: Optimization can be cancelled at any time, and the current value being displayed at that moment is then used.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:CFACtor:MODE` on page 271

Desired Crest Factor

Enters the desired crest factor.

This is only possible when the optimization "Target Crest" has been selected.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:CFACtor` on page 271

Table Setup Assistant

The "Table Setup Assistant" section can be used to set a selectable carrier range. The carrier table can be edited in the "Carrier Table" dialog.

Carrier Start

Defines the start index of the carrier range to which the following settings are intended to apply.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STARt` on page 274

Carrier Stop

Defines the stop index of the carrier range to which the following settings are intended to apply.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STOP` on page 275

Carrier State

Switches the carriers in the carrier range on/off.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STATe` on page 275

Power Start

Sets the power of the starting carrier.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:POWer[:STARt]` on page 274

Power Step

Sets the width of the step with which the power will be changed from carrier to carrier.

The carrier power that is set with **Power + n* Power Step** must be within the valid value range -80 dB to 0 dB.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:POWer:STEP` on page 274

Phase Start
Sets the phase of the starting carrier. This setting is only available for "Optimize Crest Factor Mode" = Off.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:PHASe[:STARt]` on page 273

Phase Step
Sets the width of the step with which the phase will be changed from carrier to carrier.

The phase that is set with **Phase + n* Phase Step** must be within the valid value range 0° to 360°.

This setting is only available for "Optimize Crest Factor Mode" = Off.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:PHASe:STEP` on page 273

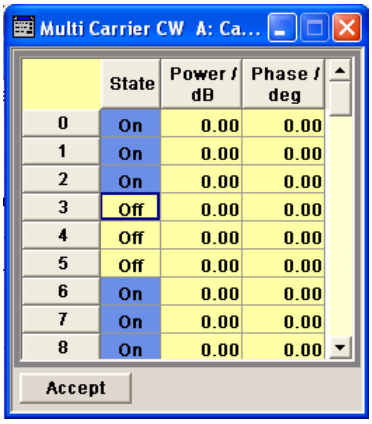
Accept
Adopts the carrier range setting into the table ("Carrier Table").

SCPI command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:EXECute` on page 272

Carrier Table and Carrier Graph

The lower part of the menu is used to open submenus for configuring carrier tables and for setting triggers, markers and clocks.

Carrier Table
Calls the table for configuring individual carriers. This configuration can be checked with the aid of the "Carrier Graph".



	State	Power / dB	Phase / deg
0	On	0.00	0.00
1	On	0.00	0.00
2	On	0.00	0.00
3	Off	0.00	0.00
4	Off	0.00	0.00
5	Off	0.00	0.00
6	On	0.00	0.00
7	On	0.00	0.00
8	On	0.00	0.00

The table displays the settings of all available carriers. Carriers in the On state are highlighted. All carrier parameters can be edited in the table.

The Multi Carrier CW signal is only computed when the "Accept" button is pressed. Whenever the table contains settings that have not yet been adopted with the "Accept" button, the background is yellow.

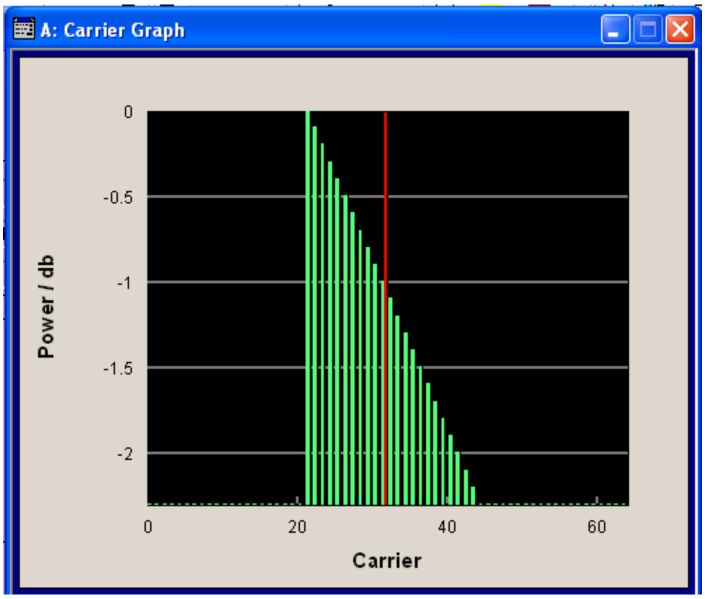
Note: The phase/deg settings are only valid if optimization of the crest factor is disabled ("Optimize Crest Factor" = Off).

"No."	This is the carrier index.
"State"	Switch a carrier on/off
"Power"	Sets the power of a carrier.
"Phase"	Sets the starting phase of a carrier.
"Accept"	Transfer the settings in the carrier table into the instrument.

SCPI command:
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:STATe on page 269
[:SOURce<hw>]:BB:MCCW:CARRier:STATe on page 271
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:POWer on page 268
[:SOURce<hw>]:BB:MCCW:CARRier:POWer on page 270
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:PHASe on page 268
[:SOURce<hw>]:BB:MCCW:CARRier:PHASe on page 269

Carrier Graph
Calls a graphical representation of the chosen carrier configuration.

The carriers are on the X-axis and the colored bars represent those carriers which are in the On state. Power is on the Y-axis, and the height of the bars corresponds to the chosen power of each individual carrier.



SCPI command:
n.a.

Marker
Calls the "Marker" dialog (see "Marker Mode" on page 90).

SCPI command:
n.a.

Marker Mode

The marker output signal for synchronizing external instruments is configured in the marker settings section "Marker Mode".

Marker Mode
Selects a marker signal for the associated "MARKER" output.

"Restart" A marker signal is generated at the start of the waveform.

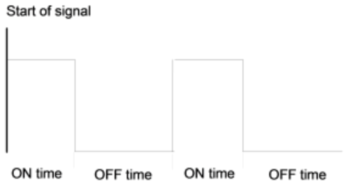
"Pulse" A regular marker signal is generated. The pulse frequency is defined by entering a divider. The frequency is derived by dividing the sample rate by the divider. The input box for the divider opens when "Pulse" is selected, and the resulting pulse frequency is displayed below it.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PULSe:DIVider` on page 277
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PULSe:FREQuency` on page 277

"Pattern " A marker signal that is defined by a bit pattern is generated. The pattern has a maximum length of 64 bits and is defined in an input field which opens when pattern is selected.

SCPI command:
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PATTern` on page 276

"ON/OFF Period" A regular marker signal that is defined by an ON/OFF ratio is generated. A period lasts one ON and OFF cycle.
The "ON Time" and "OFF Time" are each expressed as a number of samples and are set in an input field which opens when ON/OFF ratio is selected.



SCPI command:
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:ONTime` on page 276
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:OFFTime` on page 276

SCPI command:
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:MODE` on page 275

5.10.4 Multi Carrier Waveforms

Multi carrier waveforms consisting of up to 32 carriers modulated by user-selectable baseband signals can be created in order to simulate complex multi carrier scenarios with different baseband signals (e.g. CDMA2000 or 3GPP FDD).

The carriers are equally spaced and centered toward the RF frequency or baseband DC line, respectively. The carrier spacing is adjustable within the total available baseband bandwidth of 80 MHz. Each carrier can be separately defined in terms of power, phase and modulated input signal. Optionally, crest factor optimization can be applied.

Thus, multi carrier waveforms can be very easily configured as broadband test signals and used for such purposes as transmitter or receiver tests. Even complex multi carrier scenarios composed of signals from different digital standards can be created and used for these tests.

Example for the creation of a multi carrier waveform file

1. Configure general settings
2. Configure carrier table
3. Enter file name of multi carrier waveform ("Output File")
4. Save and load multi carrier waveform ("Create" or "Create and Load")

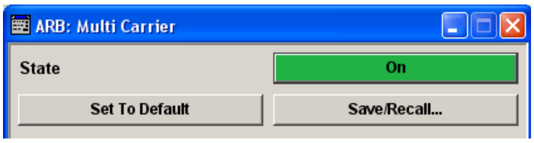
User Interface

To access the "Multi Carrier" dialog, select "Baseband > Multi Carrier...".

The menu can be used to create a multi carrier waveform and save it under its own name. As with normal waveforms, the file extension is `*.wv`. Information on clock rate, number of samples and creation day is displayed in the File menu when a waveform is loaded.

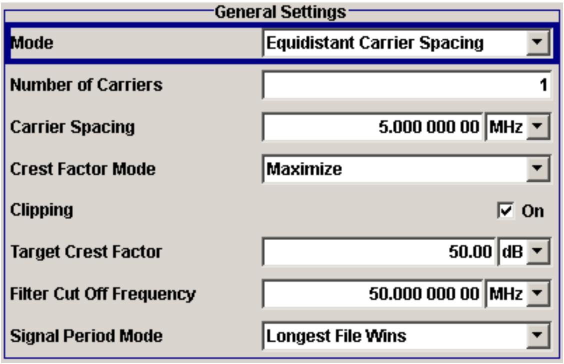
The "Multi Carrier" dialog is divided into the following sections.

The upper part of the menu is used for calling the default settings and saving and recalling the submenu settings.



State: On

The General Settings section is used to configure the main multi carrier waveform.

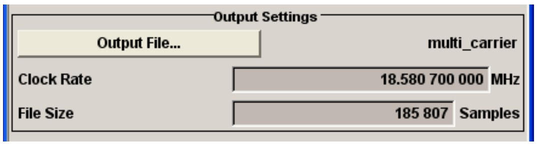


General Settings

Mode	Equidistant Carrier Spacing
Number of Carriers	1
Carrier Spacing	5.000 000 00 MHz
Crest Factor Mode	Maximize
Clipping	☑ On
Target Crest Factor	50.00 dB
Filter Cut Off Frequency	50.000 000 00 MHz
Signal Period Mode	Longest File Wins

The buttons in the middle part of the menu open submenus for defining the carrier table. The carrier settings can be checked in the graphical "Carrier Graph" submenu.

The "Output Settings" section indicates the name, the size and the clock rate of the multi carrier waveform (after calculation).



Output Settings

Output File...	multi_carrier
Clock Rate	18.580 700 000 MHz
File Size	185 807 Samples

The buttons in the lower part of the menu activate creation and optionally loading of multi carrier waveform files into the main ARB menu.

5.10.4.1 Default Settings and State

The upper part of the dialog is used for calling the default settings and saving and recalling existing ARB multi carrier submenu settings.

State Multi Carrier
Enables/disables multi carrier generation.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:STATe` on page 220

Set to Default
Calls default settings. The values are shown in the following table.

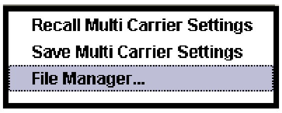
Parameter	Value
Number of Carriers	1
Carrier Spacing	0 MHz

Parameter	Value
Crest Factor Mode	Off
Signal Period Mode	Longest file wins

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:PRESet` on page 218

Save/Recall Frame
Calls the "Save/Recall" menu.

From the "Save/Recall" menu the "File Select" windows for saving and recalling the configurations of the ARB "Multi Carrier" submenu and the "File Manager" can be called.



Multi Carrier configurations are stored as files with the predefined file extension `*.arb_multcarr`. The file name and the directory they are stored in are user-definable.

The complete settings in the Multi Carrier menu are saved and recalled.

"Recall Multi Carrier Settings" Opens the "File Select" window for loading a saved Multi Carrier configuration.
The configuration of the selected (highlighted) file is loaded by pressing the "Select" button.

"Save Multi Carrier Settings" Opens the "File Select" window for saving the current Multi Carrier signal configuration.
The name of the file is specified in the "File name" entry field, the directory selected in the "Save Into" field. The file is saved by pressing the "Save" button.
The "Fast Save" checkbox determines whether the instrument performs an absolute or a differential storing of the settings. Enable this function to accelerate the saving process by saving only the settings with values different to the default ones. "Fast Save" is not affected by the "Preset" function.

"File Manager" Calls the "File Manager".
The "File Manager" is used to copy, delete, and rename files and to create new directories.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:CATalog` on page 219
`[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:LOAD` on page 219
`[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:STORe:FAST` on page 219
`[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:STORe` on page 219

5.10.4.2 General Settings

The "General Settings" section is used to configure the Multi Carrier signal.

Mode

Selects the carrier frequency mode.

"Equidistant Carrier Spacing"	Sets an equidistant carrrier spacing. The carrier frequency in the carrier table is not configurable.
"Arbitrary Carrier Frequency"	Enables you to specify the carrier frequency in the carrier table. Carrier spacing is irrelevant.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier:MODE` on page 208

Number of Carriers

Sets the number of carriers for the multi carrier waveform.

By default the multi carrier table lists 1 carrier. A maximum of 32 carriers can be configured and activated.

When the number of carriers is increased, the multi carrier table is extended by adding further lines at the end of the table. If these carrier already have been defined before, the settings are preset according to the former settings. Otherwise the parameters are preset with the default values.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier:COUNt` on page 208

Carrier Spacing

Sets the frequency spacing between adjacent carriers of the multi carrier waveform.

The carriers are arranged symmetrically around the RF carrier.

The maximum carrier spacing is limited to

Carrier spacing = Total baseband bandwidth /(Number of carriers - 1);

The total baseband bandwidth is 80 MHz.

Note: In order to avoid wrap-around problems, the effective "Carrier Spacing" might be slightly modified.
The "Carrier Spacing" is rounded in that way that the carrier closest to the baseband DC line shows no phase jump assuming that the carrier is unmodulated.

- For odd number of carriers:
  RoundedCarrierSpacing=1/OutputSignalDuration* round(CarrierSpacing * OutputSignalDuration);
- For even number of carriers:
  RoundedCarrierSpacing=2/OutputSignalDuration*round(0.5 *CarrierSpacing * OutputSignalDuration).

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier:SPACing` on page 209

Crest Factor Mode

Selects the mode for optimizing the crest factor by calculating the carrier phases.

The crest factor represents the ratio of the peak voltage value to the rms voltage value. The higher the crest factor and resulting dynamics of a signal, the greater the requirement for a power amplifier fed by the signal to be linear.

The following modes are available:

"Off"	There is no automatic setting for minimizing or maximizing the crest factor. The "Phase" setting as defined in the carrier table is in use.
"Minimize"	The crest factor is minimized by internally calculating optimized carrier phases. The Phase setting displayed in the carrier table is invalid.
"Maximize"	The crest factor is maximized by internally calculating optimized carrier phases. The Phase setting displayed in the carrier table is invalid.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CFACtor:MODE` on page 211

Clipping

Switches baseband clipping on and off.

Clipping reduces the peak power of the resulting multi carrier signal according to the input parameter "Target Crest Factor".

The resulting clipped peak power is defined by sum of the the RMS level of the unclipped multi carrier signal and the input parameter "Target Crest Factor". Note that clipping reduces also the RMS level. Hence the resulting crest factor of the clipped signal is slightly above the "Target Crest Factor". In order to get the unclipped parts of the clipped multi-carrier signal matched with the unclipped multicarrier signal, the RF output power should be reduced by the difference between resulting crest factor and "Target Crest Factor".

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CLIPping[:STATe]` on page 213

Target Crest Factor

Sets the value of the desired crest factor, if baseband clipping is enabled.

A "Target Crest Factor" above the crest factor of the unclipped multicarrier signal has no effect.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CLIPping:CFACtor` on page 212

Filter Cut Off Frequency

Sets the cut off frequency of the final lowpass filter, if baseband clipping is enabled.

When the cut off frequency is set as half of the output sample rate, a final lowpass filter improves the spectrum of the clipped multicarrier signal, but may also increase the resulting crest factor.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CLIPping:CUToff` on page 212

Signal Period Mode

Selects the mode for calculating the resulting signal period of the multi carrier waveform. The carrier table provides an information button to obtain sample rate and file length data of each carrier.

The resulting period is always calculated for all carriers in the carrier table irrespective of their state (ON/OFF).

Note: Wrap-around and timing problems may occur when I/Q signals of different length are used. Thus, demodulation of a carrier may be difficult or even impossible. It is therefore recommended to consider the timing already when creating the input I/Q files or to adjust the signal duration to the carrier which is subsequently demodulated (in this case, the other carriers are for interfering the signal only). These problems do not arise with signals of the same standard (e.g. 3GPP).

The following modes are available:

"Longest File Wins" The resulting signal period is defined by the longest I/Q file in the carrier table. Shorter I/Q files are periodically repeated.

"Shortest File Wins" The resulting signal period is defined by the shortest I/Q file in the carrier table. Only the first part of longer I/Q files is used.

"User" The signal period can be set by the user in the "Signal Period" field. Shorter I/Q files are repeated periodically, and only the first part of longer I/Q files is used.

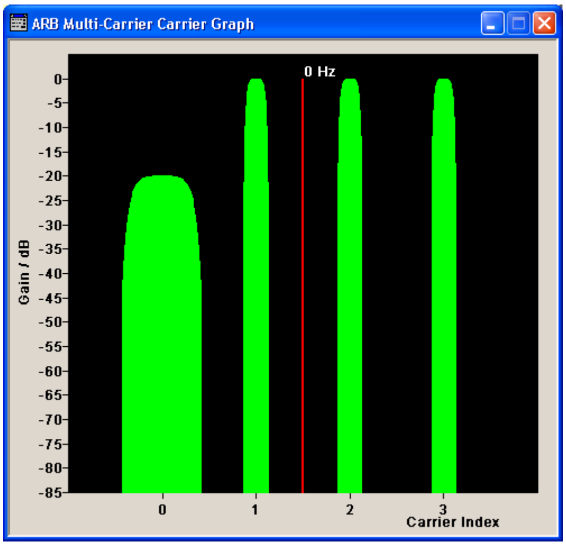
SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:TIME:MODE` on page 220

Carrier Table Submenu
Calls the table for configuring individual carriers. A selectable carrier range can be set with the aid of the optional "Carrier Table Assistant" (see chapter 5.10.4.4, "Muti Carrier Table", on page 98).

The multi carrier configuration can be checked with the aid of the "Carrier Graph".

Carrier Graph
Calls a graphical representation of the current multi carrier configuration in the frequency domain.

The height of the bars corresponds to the chosen gain of each individual carrier. The bandwidth of the carriers signals is indicated by the width of the bars.



5.10.4.3 Output Settings

The "Output Settings" section in the multi carrier main menu indicates the name, the size and the clock rate of the currently calculated multi carrier output file.

File

Opens the "File" menu, where the output file name of the multi carrier waveform which has to be calculated can be entered.

The multi carrier waveform is calculated and saved under this name by clicking the "Create" or "Create and Load" button.

SCPI command:

`[:SOURce<hw>]:BB:ARB:MCARrier:OFILe` on page 218

Clock Rate

Displays the resulting sample rate at which the multi carrier waveform is output by the arbitrary waveform generator. The output clock rate depends on the number of carriers, carrier spacing and input sample rate of the leftmost or rightmost carriers.

SCPI command:

`[:SOURce<hw>]:BB:ARB:MCARrier:CLOCk` on page 214

File Size

Displays the resulting number of samples of the multi carrier waveform.

SCPI command:

`[:SOURce<hw>]:BB:ARB:MCARrier:SAMPles` on page 218

Create

Creates a new multi carrier waveform defined by the multi carrier table and General Setting entries.

This multi carrier waveform is saved with the file name specified in "Output File". If no file name is specified, the File Manager opens so that the file name can be entered. As with normal waveforms, the file extension is `*.wv`.

Depending on the configuration of the multi carrier waveform, calculation may take some time. A panel with a progress bar and an "Abort" button appears during the calculation process.

SCPI command:

`[:SOURce<hw>]:BB:ARB:MCARrier:CREate` on page 214

Create and Load

Creates a new multi carrier waveform defined by the multi carrier table and General Setting entries and loads it subsequently in the "ARB" main menu.

This multi carrier waveform is saved with the file name specified in "Output File". If no file name is specified, the File Manager opens so that the file name can be entered. As with normal waveforms, the file extension is `*.wv`.

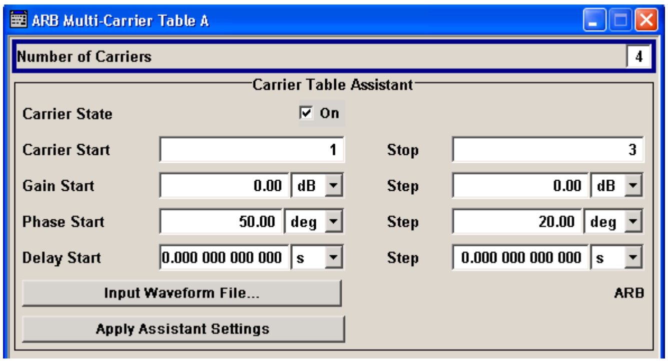
Depending on the configuration of the multi carrier waveform, calculation may take some time. A panel with a progress bar and an "Abort" button appears during the calculation process.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CLOad` on page 213

5.10.4.4 Muti Carrier Table

The "Multi Carrier Table" manu is oped by means of the "Carrier Table" button in the middle part of the Multi Carrier Waveforms menu. The carrier settings can be checked in the graphical "Carrier Graph" submenu.

The "Carrier Table Assistant" section can be used to set a selectable carrier range. This section serves as a optional mean to quickly set up a multi carrier scenario within a specified carrier range.



ARB Multi-Carrier Table A

Number of Carriers 4

Carrier Table Assistant

Carrier State ☑ On

Carrier Start 1 Stop 3

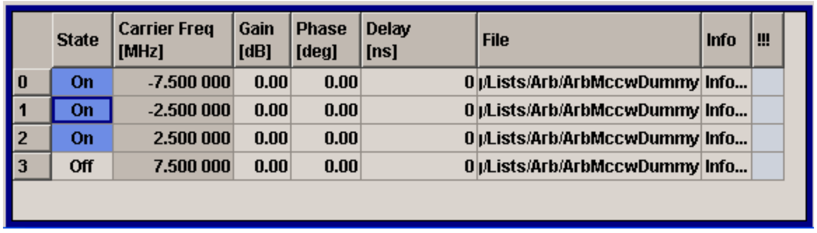
Gain Start 0.00 dB Step 0.00 dB

Phase Start 50.00 deg Step 20.00 deg

Delay Start 0.000 000 000 000 s Step 0.000 000 000 000 s

Input Waveform File... ARB

Apply Assistant Settings



	State	Carrier Freq [MHz]	Gain [dB]	Phase [deg]	Delay [ns]	File	Info	!!!
0	On	-7.500 000	0.00	0.00	0	/Lists/Arb/ArbMccwDummy	Info...	
1	On	-2.500 000	0.00	0.00	0	/Lists/Arb/ArbMccwDummy	Info...	
2	On	2.500 000	0.00	0.00	0	/Lists/Arb/ArbMccwDummy	Info...	
3	Off	7.500 000	0.00	0.00	0	/Lists/Arb/ArbMccwDummy	Info...	

Carrier Table Assistant

Number of Carriers

Defines the number of carriers of the multi carrier waveform.

This parameter is identical to that in the "General Setting" section.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier:COUNt` on page 208

Carrier State

Switches the carriers in the range "Carrier Start" to "Carrier Stop" on/off.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:STATe` on page 218

Carrier Start

Defines the start index of the carrier range to which the assistant settings are intended to apply.

SCPI command:

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:STARt on page 217

Carrier Stop

Defines the stop index of the carrier range to which the assistant settings are intended to apply.

SCPI command:

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:STOP on page 217

Gain Start

Sets the gain of the carrier marked by "Carrier Start".

SCPI command:

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:POWer[:STARt] on page 217

Gain Step

Sets the step width that is used to increment the gain

The resulting carrier gain in the carrier table equals:

GainStart + n* Gain Step

where n ranges from 0 to (Carrier Stop - Carrier Start).

SCPI command:

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:POWer:STEP on page 217

Phase Start

Sets the phase of the carrier marked by "Carrier Start".

SCPI command:

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:PHASe[:STARt] on page 216

Phase Step

Sets the step width that is used to increment the phase.

The resulting phase in the carrier table equals:

PhaseStart + n* PhaseStep

where n ranges from 0 to (Carrier Stop – Carrier Start)

SCPI command:

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:POWer:STEP on page 217

Delay Start

Sets the delay of the carrier marked by "Carrier Start".

SCPI command:

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:DELay[:STARt] on page 215

Delay Step

Sets the step width that is used to increment the delay

The resulting delay in the carrier table equals:

DelayStart + n* DelayStep

where n ranges from 0 to (Carrier Stop – Carrier Start).

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:DELay:STEP` on page 214

Input Waveform File

Calls the "File" menu for selecting the inputfile with the I/Q signal to be modulated onto all carriers of the selected carrier range.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:FILE` on page 216

Apply Assistant Settings

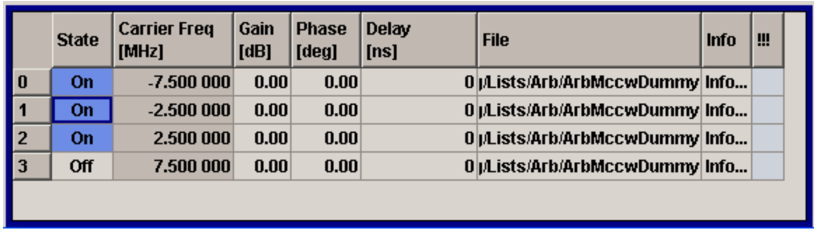
Transfer the assistant settings to the carrier table.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:EXECute` on page 215

Carrier Table

The table displays the settings of all available carriers. Previously applied assistant settings can be further refined. The number of lines corresponds to the number of carriers.

Note: The phase/deg settings are only valid if optimization of the crest factor is disabled ("Crest Factor Mode" = Off).



	State	Carrier Freq [MHz]	Gain [dB]	Phase [deg]	Delay [ns]	File	Info	!!!
0	On	-7.500 000	0.00	0.00	0	ı/Lists/Arb/ArbMccwDummy	Info...	
1	On	-2.500 000	0.00	0.00	0	ı/Lists/Arb/ArbMccwDummy	Info...	
2	On	2.500 000	0.00	0.00	0	ı/Lists/Arb/ArbMccwDummy	Info...	
3	Off	7.500 000	0.00	0.00	0	ı/Lists/Arb/ArbMccwDummy	Info...	

No ← Carrier Table

Indicates the carrier index ranging from 0 to (number of carriers -1).

Individual carriers can be set using the remote control commands by specifying the index in the parameter CARR.

State ← Carrier Table

Switches On/Off a carrier.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:STATe` on page 211

Carrier Freq [MHz] ← Carrier Table

Sets the carrier frequency.

Note: The carrier frequnecy can only be set in "Arbitrary Carrier frequency" mode.

For "Equidistant Carrier Spacing", the carrier spacing is determined automatically.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:FREQuency` on page 210

Gain ← Carrier Table
Sets the gain of a carrier.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:POWer` on page 211

Phase ← Carrier Table
Sets the starting phase of a carrier.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:PHASe` on page 211

Delay ← Carrier Table
Sets the starting delay of a carrier.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:DELay` on page 210

File ← Carrier Table
Calls the "File" menu for selecting the input file with the I/Q signal to be modulated onto the carrier.

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:FILE` on page 210

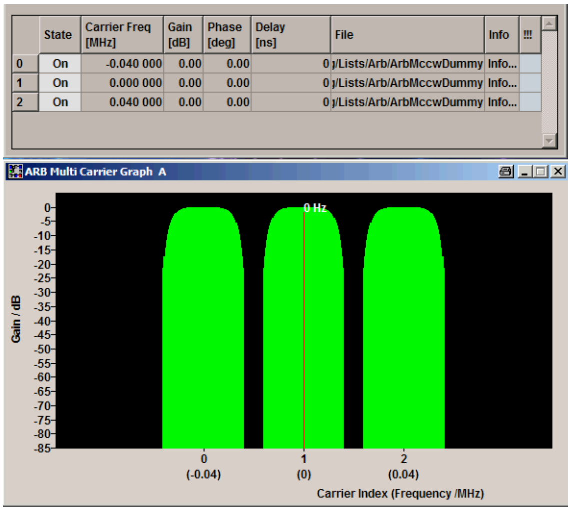
Info ← Carrier Table
Indicates the sample rate, number of I/Q value pairs (number of samples), and the resulting signal period of the selected I/Q input file.

SCPI command:
n.a.

!!! ← Carrier Table
Indicates an occurred conflict by a means of a red LED. A conflict arises when the carriers overlap.

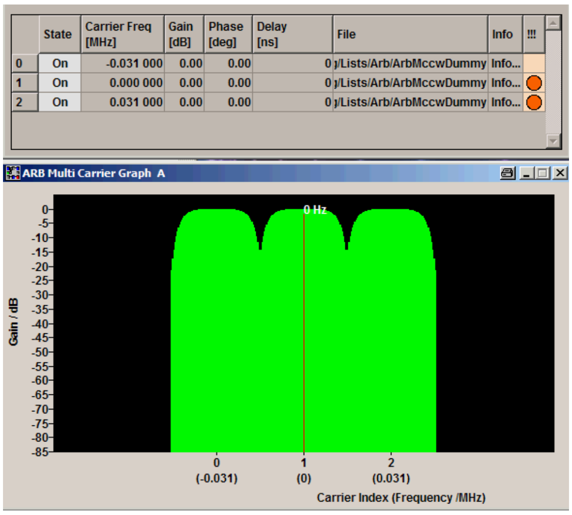
Example:

A conflict is indicated both in the table, as well as in the graph, as shown in the following figures:



	State	Carrier Freq [MHz]	Gain [dB]	Phase [deg]	Delay [ns]	File	Info	!!!
0	On	-0.040 000	0.00	0.00	0	y/Lists/Arb/ArbMccwDummy	Info...	
1	On	0.000 000	0.00	0.00	0	y/Lists/Arb/ArbMccwDummy	Info...	
2	On	0.040 000	0.00	0.00	0	y/Lists/Arb/ArbMccwDummy	Info...	

example without carrier conflicts



	State	Carrier Freq [MHz]	Gain [dB]	Phase [deg]	Delay [ns]	File	Info	!!!
0	On	-0.031 000	0.00	0.00	0	y/Lists/Arb/ArbMccwDummy	Info...	
1	On	0.000 000	0.00	0.00	0	y/Lists/Arb/ArbMccwDummy	Info...	●
2	On	0.031 000	0.00	0.00	0	y/Lists/Arb/ArbMccwDummy	Info...	●

example with carrier conflicts

SCPI command:
`[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:CONFlict` on page 209

5.10.4.5 Typical Applications for Multi Carrier Waveforms

High Power Amplifiers of multi carrier base stations face increased requirements in terms of linearity and acceptable intermodulation performance.

A standard transmitter test might be quickly setup by the following instructions:

1. Load a standardized 3GPP downlink test model, e.g. "Test Model 1_16" and set State to "ON".
2. Create a 3GPP Testmodel ARB file by activating "Generate Waveform File".
3. Setup a multi carrier scenario with 4 carriers and a carrier spacing of 5 MHz.
4. Apply the input file containing the previously generated 3GPP testmodel to all 4 carriers.

5.10.5 Multi Segment Waveforms

To fulfill the requirements of some test systems and to enable rapid alternation between different waveforms with differing test signals, the R&S Signal Generator provides the functionality to generate multi segment waveform files. A multi segment waveform is a combination of multiple waveforms called segments. Each segment represents a completely independent waveform that can be output with its own marker and clock settings. The entire multi segment waveform is loaded into memory. The segment intended to be output at any given moment can be selected by the user. It is therefore possible to alternate between the individual waveform segments without experiencing any delay due to the loading operation.

If very high switchover speeds are required, the test signals can be continuously scrolled through with the aid of an external trigger or by applying a predefined “play list”. For this purpose the segments must have a common sample rate. If the combined waveforms have different sample rates, they can be adapted to a common sample rate by resampling. By the same procedure the instantaneous amplitude of the various waveforms can be scaled to a common RMS level.

To access the "Multi Segment" dialog, select "Baseband > Multi Segment...".

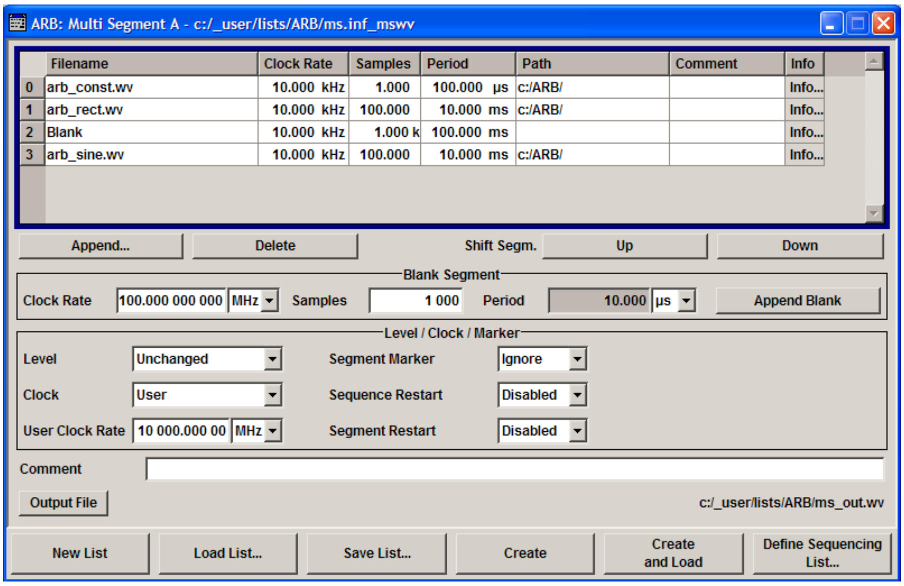
The menu can be used to create a multi segment waveform from existing waveforms and save it under its own name. As with normal waveforms, the file extension is `*.wv`. Information on whether a file is a multi segment waveform is displayed in the File menu with the tag information when a waveform is loaded. Additionally, an information about the waveform can be aquired in the "Waveform Info" dialog.

If the combined waveforms have different sample rates, they can be adapted to a common sample rate by resampling. By the same procedure the instantaneous amplitude of the various waveforms can be scaled to a common rms level.

The configuration of a multi segment waveform, that is to say details of how it is made up from different waveforms, the level and clock rate settings, and the file name, can be saved separately in a list. The file extension is `*.inf_mswv`. This method can be used to create any number of configurations as a basis for defining further multi segment waveforms.

Example for the creation of a multi segment waveform file.

1. Select "New List" to create empty list.
2. Use the "Append" function to add two or more waveform files.
3. Adjust the "Level", "Clock" and "Segment Marker" settings.
4. Select the "Output File" and enter file name.
5. Select "Save List" to save the configuration.
6. Change the "State" to On to enable the generation of the multi segment waveform.



ARB: Multi Segment A - c:/_user/lists/ARB/ms.inf_mswv

	Filename	Clock Rate	Samples	Period	Path	Comment	Info
0	arb_const.wv	10.000 kHz	1.000	100.000 µs	c:/ARB/		Info...
1	arb_rect.wv	10.000 kHz	100.000	10.000 ms	c:/ARB/		Info...
2	Blank	10.000 kHz	1.000 k	100.000 ms			Info...
3	arb_sine.wv	10.000 kHz	100.000	10.000 ms	c:/ARB/		Info...

Blank Segment

Clock Rate 100.000 000 000 MHz Samples 1 000 Period 10.000 µs

Level / Clock / Marker

Level Unchanged
Clock User
User Clock Rate 10 000.000 00 MHz
Segment Marker Ignore
Sequence Restart Disabled
Segment Restart Disabled

Comment

Output File c:/_user/lists/ARB/ms_out.wv

5.10.5.1 Multi Segment File Contents

The upper part of the dialog provides the functions, necessary to create and adjust the contents of the multi segment file.

State

Enables/disables multi segment generation.

SCPI command:

`[:SOURce<hw>]:BB:ARB:WSEGment:STATe` on page 229

Multi Segment Table

The table lists the individual waveforms (segments) of the selected multi segment waveform. The information about the segments is taken from the tags of the corresponding waveform files.

"Segment#" Indication of segment index. The segment index is used to select the segment to be output.
The segment index also defines the sequence of the output during triggering in the "Next Segment" or "Next Segment Seamless".

"Waveform" Indication of the waveform file name of the segment.

"Clock Rate" Indication of the clock rate of the segment.

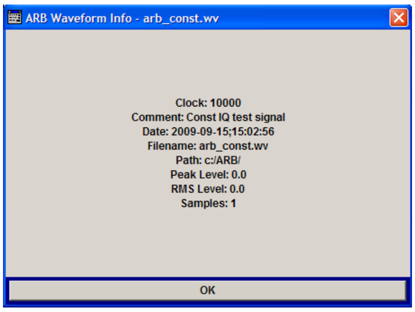
"Samples" Indication of the number of samples in the segment.

"Period" Indication of the segment duration.

"Path" Indication of the location of the waveform file used for the corresponding segment.

"Comment" Indication of the comment on the segment.

"Info" Opens a dialog with detailed information about the currently selected waveform.



Clock: 10000
Comment: Const IQ test signal
Date: 2009-09-15;15:02:56
Filename: arb_const.wv
Path: c:/ARB/
Peak Level: 0.0
RMS Level: 0.0
Samples: 1

SCPI command:
n.a.

Append

Opens the file menu to enter the file name of the waveform file to be appended.

SCPI command:

`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:SEGMent:CATalog` on page 226

`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:SEGMent:APPend` on page 225

Delete

Deletes the selected segment.

SCPI command:

`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:DELete` on page 223

Shift Seg.# Up/Down

Shifts the selected segment up and down.

5.10.5.2 Blank Segment

The Blank Segment section is where a blank segment can be configured.

Clock Rate

Selects the clock rate of the blank segment.

SCPI command:

`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:BLANk:APPend` on page 221

Samples

Selects the number of samples for the blank segment.

SCPI command:

`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:BLANk:APPend` on page 221

Period

Displays the resulting period for the blank segment.

SCPI command:
n.a.

Append Blank

Adds the blank segment to the multi segment file.

SCPI command:
`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:BLANk:APPend` on page 221

5.10.5.3 Level / Clock /Marker

The Level/Marker/Clock section provides the parameters necessary to adjust the level, marker and clock settings for the selected multi segment waveform file.

Level

Defines the level mode for the multi segment waveform.

"Unchanged" Conserning the level settings, the segments are output exactly as defined in the files. The "Level" display applies only to the segment with the highest rms value. In some circumstances the remaining segments are output at a lower level than that displayed.

"Equal RMS" Segments are output so that all segments have the same rms value. The "Level" display applies to all segments.

SCPI command:
`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:LEVel[:MODE]` on page 223

Clock

Selects the clock rate mode for the multi segment waveform.

"Unchanged" A segment is output with the clock rate defined in the file.
If segments have different clock rates, extended trigger mode "Next Segment" allows internal segment switchovers only ("Internal" trigger source).
Extended trigger mode "Next Segment Seamless" can only be selected if all segments have the same clock rate.

"Highest" All segments are output at the highest available clock rate.
This mode provides very short switchover times between segments.
The time for calculating the multi segment waveform is increased since the individual segments have to be resampled.

"User" All segments are output at the clock rate defined in "User Clock".
This mode provides very short switchover times between segments.
The time for calculating the multi segment waveform is increased since the individual segments have to be resampled.

SCPI command:
`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:CLOCk:MODE` on page 222

User Clock Rate

Defines the sample rate used for multi segment waveform output in case of Clock Mode "User".

SCPI command:

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:CLOCk on page 222

Segment Marker

Defines the way the marker information within the separate segments is processed.

"Ignore" The marker information carried in the individual segment waveform files is not considered.

"Take Over" The output waveform file contains the marker information as configured in the individual waveform files.

SCPI command:

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:MARKer:MODE on page 224

Sequence Restart

Enables/disables the generation of an additional marker restart signal in the output waveform file.

If additional marker generation is enabled, the existing marker signals in the individual segment waveform files are not considered.

"Disable" No additional marker is generated.

"Marker 1/2/3/4" Generates a restart marker signal at the beginning of the first segment for the corresponding marker.
Use this setting to generate a restart marker for the complete sequence.

SCPI command:

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:MARKer:FSEGment on page 224

Segment Restart

Enables/disables the generation of an additional marker restart signal in the output waveform file.

If additional marker generation is enabled, the existing marker signals in the individual segment waveform files are not considered.

"Disable" No additional marker is generated.

"Marker 1/2/3/4" Generates a restart marker signal at the beginning of each segment for the corresponding marker.
The segment begin is defined by the low-high slope of the marker. This applies for switching between two segments as well as in case of segment replay.

SCPI command:

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:MARKer:ESEGment on page 223

5.10.5.4 Output File

In the Output File section, the output file name is specified and a comment can be added to the file.

Comment
Adds a comment to the multi segment file.

SCPI command:
`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:COMMent` on page 223

Output file
Opens the File menu, where the file name of the multi segment waveform which has to be calculated can be entered. The multi segment waveform is saved under this name by clicking the "Create" or "Create and Load" button.

A name must also be entered here before the list can be saved as a configuration file ("Save List"). In this case the name is needed for internal storage procedures. It is entered as the default name for the multi segment waveform file when loading the list.

SCPI command:
`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:OFILe` on page 225

5.10.5.5 Handling of Multi Segment Files

The buttons in this sections are provided for handling of the multi segment files.

Save List
Saves the current entries of the table in a configuration file, including the level mode, clock mode, segment marker mode and output file name settings.

The file name is entered in the "File" menu. Configuration files have the file extension `*.inf_mswv`. They can be used later as the basis for further multi segment waveforms.

Before a multi segment configuration file can be saved, a file name must be entered in "Output File", since this is saved with the configuration file for internal purposes.

SCPI command:
The configuration file to be used for the creation of the multi segment waveform is defined with the command:
`[:SOURce<hw>]:BB:ARB:WSEGment:CREate`
The file name of the waveform file is always determined with the command `[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:OFILe`

Load List
Opens the file menu to select the configuration file to be edited.

SCPI command:
`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:CATalog` on page 221
`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:SELect` on page 226

New List
Calls the "Create Multi Segment Waveform List" dialog to enter the name of the new file.

A new new configuration table is created.

SCPI command:
`[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:SELect` on page 226

Define Sequencing List
Open a dialog box for configuring of the play lists (see chapter 5.10.5.6, "Multi Segment Waveform Sequencing", on page 109).

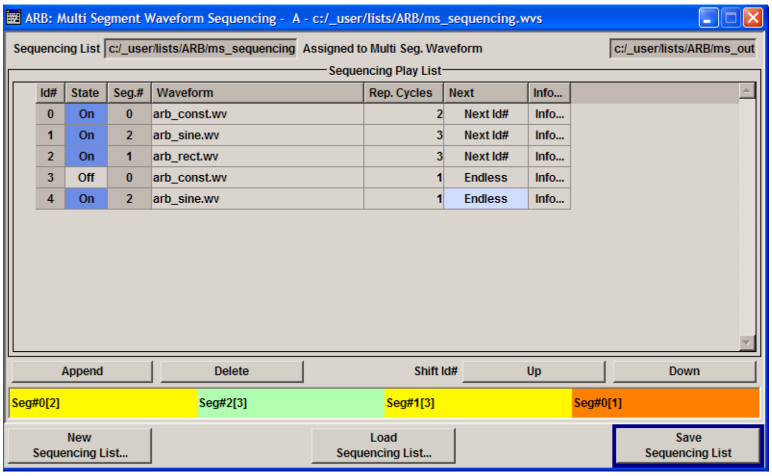
SCPI command:
n.a.

5.10.5.6 Multi Segment Waveform Sequencing

The settings provided in the "Multi Segment Waveform Sequencing" dialog can be adjusted to crate a “play list”, i.e. to define the sequence in which the waveform file segments are processed and how many times each of them is repeated. The configured play list is displayed in form of a "Sequencing Graphic".

All segments in a “play list” must have the same clock rate.

In the "ARB" main dialog or in the "ARB Trigger/Marker/Clock" dialog, set the "Next Segment Mode" to "Sequencer" to activate the configured “play list”.



ARB: Multi Segment Waveform Sequencing - A - c:/_user/lists/ARB/ms_sequencing.wvs

Sequencing List: c:/_user/lists/ARB/ms_sequencing
Assigned to Multi Seg. Waveform: c:/_user/lists/ARB/ms_out

Sequencing Play List

Id#	State	Seg.#	Waveform	Rep. Cycles	Next	Info...
0	On	0	arb_const.wv	2	Next Id#	Info...
1	On	2	arb_sine.wv	3	Next Id#	Info...
2	On	1	arb_rect.wv	3	Next Id#	Info...
3	Off	0	arb_const.wv	1	Endless	Info...
4	On	2	arb_sine.wv	1	Endless	Info...

Sequencing List
Displays the name of the selected sequencing list file, i.e. the file with the “play list” settings.

The sequencing list file has an extension `*.wvs` and is automatically assigned to but independent from the multi segment file. By default, both files have the same name and are located in the same file directory. More than one sequencing list files can be created per multi segment waveform file.

The sequencing list file carries information only about the segment number, the corresponding wavefile names are retrieved form the assigned multi segment waveform file, i.e. the same sequencing list file can be reused for different multi segment waveform files with the same number of segments.

Changes and re-calculations of a multi segment waveform file cause a re-check whether the assigned sequencing list files are still valid. A message is displayed to inform about necessary corrections in the “play list”.

SCPI command:
n.a.

Assigned to Multi Seg. Waveform
Displays the name of the multi segment waveform file the current sequencing list file is assigned to.

SCPI command:
n.a.

Sequencing Play List
The "Sequencing Play List" table is used to define the sequence in which the waveform file segments are processed and how many times each of them is repeated.

Id# ← Sequencing Play List
Indication of the row number.

n.a.

State ← Sequencing Play List
Enables/disables the selected row. Only active segments are processed.

n.a.

Segment# ← Sequencing Play List
Indication of segment index.

The sequencing list file carries information only about the segment index, the corresponding wavefile names are retrieved form the assigned multi segment waveform file.

n.a.

Waveform ← Sequencing Play List
Indication of the waveform file name of the segment. The file name is retrieved form the assigned multi segment waveform file.

Available for selection are only waveform files from the pool of segments of the selected multi segment waveform file.

n.a.

Repetition Cycles ← Sequencing Play List
Sets the number of times the selected segment is cyclically repeated before the next segment in the sequence is processed.

n.a.

Next ← Sequencing Play List
Determines the action after completing the current one, like for instance which segment is processed after the processing of the current one is finished.

"Next Id#" The next active segment in the play list is processed.

"Blank" After the processing of the current segment is completed, the signal output is stopped until a signal restart like a retrigger signal is received. A restart signal causes a complete restart of the sequencing play list.

"Endless" The current segment is re-played until a signal restart like a retrigger signal is received and the complete sequencing play list is restarted. Only active segments are considered.

"Goto Id#" Determines the row number of the segment to be processed next. This feature is currently limited for the first 32 rows.

SCPI command:
n.a.

Info ← Sequencing Play List
Opens a dialog with detailed information about the currently selected waveform.

SCPI command:
n.a.

Append
Inserts a new row at the end of the sequencing play list table.

SCPI command:
`[:SOURce<hw>]:BB:ARB:WSEGment:SEQuence:SELect` on page 228
`[:SOURce<hw>]:BB:ARB:WSEGment:SEQuence:APPend` on page 228

Delete
Deletes the selected row.

SCPI command:
n.a.

Shift Id# Up/Down
Shifts the selected row up and down.

SCPI command:
n.a.

Sequencing Graphic
Displays the play list graph according to the current configuration.

The graph shows the segment sequence, the segment number and the number of times, the segment is repeated; considered are only the active segments.

SCPI command:
n.a.

New Sequencing List
Opens the "Select Sequencing List" dialog to enter the name of the new file.

SCPI command:
n.a.

Load Sequencing List
Opens the "Select Sequencing List" dialog to select an existing play list file.

SCPI command:
n.a.

Save Sequencing List
Saves the current entries of the table in a play list file.

The default file name of the play list file is the same as the multi segment waveform file name, but it can be changed.

Play list files have the file extension `*.wvs`.

SCPI command:
n.a.

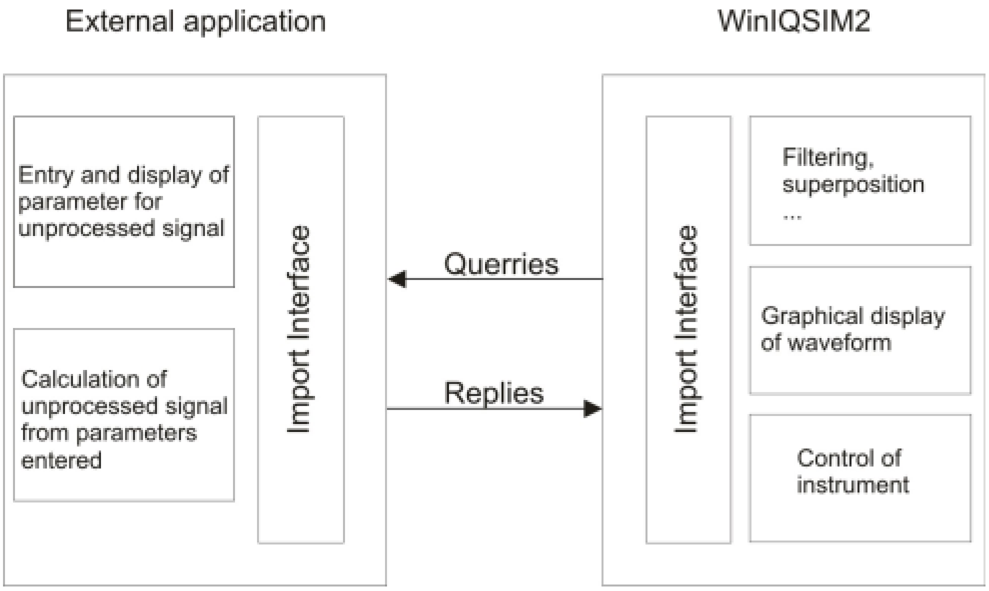
5.10.6 Import IQ Data

The import system makes it possible to extend R&S WinIQSIM2 by user-specific systems.

A second program running at the same time as R&S WinIQSIM2 transfers the waveform (unprocessed signal) to R&S WinIQSIM2. In R&S WinIQSIM2, the signal undergoes all known processing stages from filtering to signal transfer to the selected signal or modulation generator, such as R&S SMU for instance.

So, all the user has to do is generate the (unfiltered) waveform. All other functions such as filtering, graphics display, and transferring the signal to the selected instrument are provided by R&S WinIQSIM2.

The communication between R&S WinIQSIM2 and an application take place via an TCP/IP interface.


External application
WinIQSIM2
Entry and display of parameter for unprocessed signal
Calculation of unprocessed signal from parameters entered
Import Interface
Querries
Replies
Import Interface
Filtering, superposition ...
Graphical display of waveform
Control of instrument

The external application is used as a server for the data and is referred to as "import server" in the following. In certain operating states, R&S WinIQSIM2 outputs queries to the data server to which the import server replies.

The following queries are defined:

- Server identity
- Parameter status
- Sequence length
- Sequence length
- Waveform

To be able to use the TCP/IP interface, the TCP/IP driver must be installed on the PC. This is the case if, for example, the processor has a connection to a LAN or to the internet.

The advantage of the TCP/IP link is that the server can run on other processors as R&S WinIQSIM2 and use any operating system, e. g. LINUX.

TCP/IP is supported by all known development systems.

The communication between server and client can be described by means of the parameters **ServerName** (= IP Address) and **PortNumber**.

5.10.6.1 Description of TCP/IP Import Interface

The following sections apply to programming under National LabWindows CVI 5.0. The commands have similar designations in other development environments. Consult the manuals of the required development environment to familiarize yourself with programming the TCP/IP.

A TCP/IP link consists of a server and one or more several clients. The server gets the data ready and transfers them to the client on request or spontaneously. The client takes the incoming data.

R&S WinIQSIM2 WinIQSIM2 is considered here as the client and the external application as server, although the other solution would also be possible.

Server Name, Port Number, and Items

To build a link to the server, the client must know the name of the server (e.g. 12.34.56.78). Also the parties must agree on a common channel, the so-called port. The default value is 1000 but any other value may be chosen. Some ports are pre-allocated (e.g. 80 for Browser) and are to be avoided.

Communications are exchanged between client and server. Communications from client and server are ASCII strings, messages from the server are in the form of binary blocks.

Table 5-11: TCP/IP Items from Client to Server

Item	Meaning
GetServerID	ID query
GetNewState	Query if new data are available
GetSequenceLength	Query of sequence length
GetSymbolRate	Query of symbol/reference point rate in Hz
GetSymbol	Query of the symbols/reference points

Format of Query Commands

Query commands are sent with *ClientTCPWrite (gHandle, request, strlen(request), 1000)*; the last parameter timeout is given in ms.

Format of Replies (Messages) of Import Server

Replies are sent with *ServerTCPWrite ()*.

The reply must get to R&S WinIQSIM2 within 10 seconds, since otherwise timeout occurs. If the calculation (including transmission) lasts a long time, it must not be triggered by a query. The server must have a data set in reserve which it returns as a response to *GetSymbols*.

To distinguish between replies to different queries, a header is placed in front of the data section of the reply:

```
typedef enum { SYMBOLRATE, SEQUENCE_LENGTH, SYMBOLS, NEWSTATE, SERVER_ID }
SERVER_MESSAGE_HEADER_TYPE;
typedef struct
{
SERVER_MESSAGE_HEADER_TYPE type;
float data;
}
SERVER_MESSAGE_TYPE;
SERVER_MESSAGE_TYPE Reply;
Reply.type = SEQUENCE_LENGTH;
Reply.data = (float)SequenceLength;
ServerTCPWrite (gHandle, (void *)&Reply, sizeof(Reply), 1000);
```

ServerID, NewState and the symbol rate are handled the same way as the sequence length.

The above format cannot be used if the symbol value is to be sent to the WinIQSIM2 because the message is more than 1 float value long.

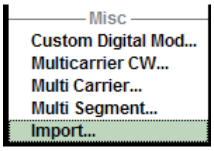
A free memory block must be filled then with data as follows:

```
Pos 1 : SYMBOLS (enum, = 2)
Pos 2 : I(0)
Pos 3 : Q(0)
Pos 4 : I(1)
Pos 5 : Q(1)
....
Pos 2*N : I(N-1)
Pos 2*N+1 : Q(N-1)
```

A pointer on this block will then be transferred to the function *ServerTCPWrite()*.

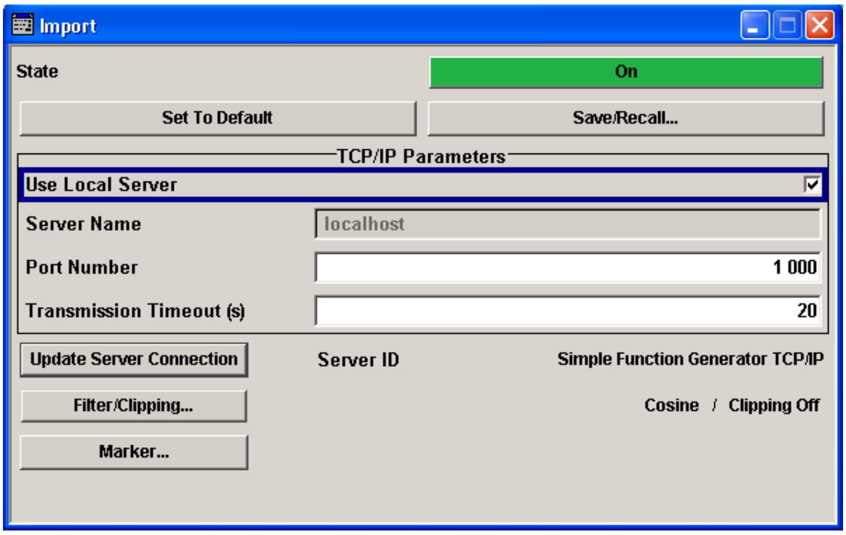
Alternatively to transmitting the waveform in one block, the samples can be divided among several blocks. In the latter case, only the first block is sent with a header, the following blocks are without header.

5.10.6.2 Import Main Menu



The menu for importing an IQ signal can be opened in the "Baseband" function block.

The Import menu allows to configure the TCP/IP connection to the server.



State: On

Use Local Server ☑

Server Name: localhost

Port Number: 1 000

Transmission Timeout (s): 20

Server ID: Simple Function Generator TCP/IP

Cosine / Clipping Off

State - Import

Enables/disables the import of IQ data.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:STATe` on page 260

Set to Default - Import

Calls the default settings. The values are shown in the following table.

Parameter	Value
Use Local Server	On
Server Name	localhost
Port Number	1000
Transmission Timeout (s)	20
Filter	Cosine
Impulse Length	Auto

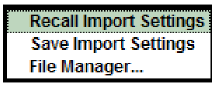
Parameter	Value
Oversampling	Auto
Clipping State	On

SCPI command:
[:SOURce<hw>]:BB:IMPort:PRESet on page 256

Save/Recall - Import

Calls the "Save/Recall" menu.

From the "Save/Recall" menu, the "File Select" windows for saving and recalling the complete settings in the "Import"menu can be called.

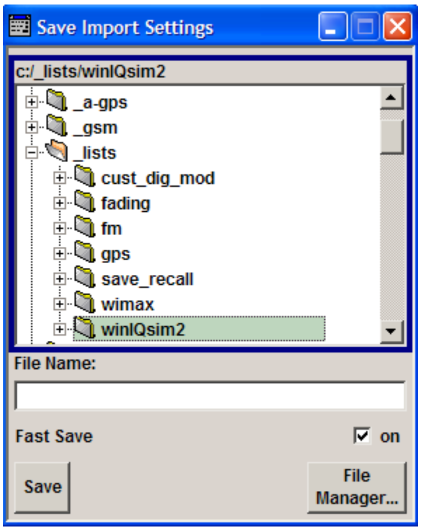


The Import settings are stored as files with the predefined file extension "*.import". The file name and the directory they are stored in are user-definable.

The complete settings in the "Import" menu are saved and recalled.

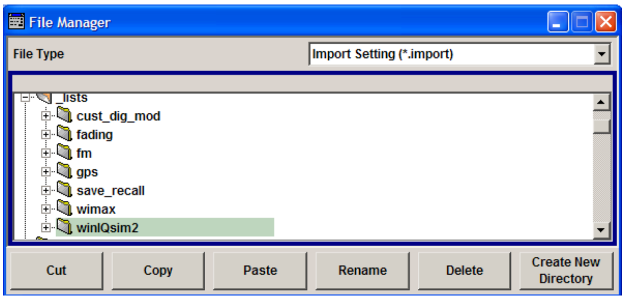
"Recall Import Settings" Opens the "Recall Import Settings"window for loading a saved import configuration.
The configuration of the selected (highlighted) file is loaded by pressing the "Select" button.

"Save Import Settings" Opens the "Save Import Settings" window for saving the current import configuration.
The name of the file is specified in the "File Name" entry field. The file is saved by pressing the "Save" button.



Fast Save ☑ on

"File Manager" Calls the "File Manager".
The "File Manager" is used to copy, delete and rename files and to create new directories.



File Type: Import Setting (*.import)

SCPI command:
[:SOURce<hw>]:BB:IMPort:SETTing:CATalog on page 258
[:SOURce<hw>]:BB:IMPort:SETTing:LOAD on page 259
[:SOURce<hw>]:BB:IMPort:SETTing:STORe on page 259
[:SOURce<hw>]:BB:IMPort:SETTing:DELete on page 258

Use Local Server - Import
Sets the local host as the server from where the IQ data streams will be dowloaded. If activated, the field for entering a remote host is filled with the entry "localhost" automatically and cannot be edited.

SCPI command:
[:SOURce<hw>]:BB:IMPort:SERVer:LOCal[:STATe] on page 257

Server Name - Import
Enters the name (IP address) of the connected server. To enter the name in this field, the check box "Use Local Server" has to be unchecked.

SCPI command:
[:SOURce<hw>]:BB:IMPort:SERVer:NAME on page 257

Port Number - Import
Enters the port number for the server connection. The default value is 1000.

SCPI command:
[:SOURce<hw>]:BB:IMPort:SERVer:PORT on page 257

Transmission Timeout (s) - Import
Sets the duration after which the transmission is aborted in case of a server timeout.

SCPI command:
[:SOURce<hw>]:BB:IMPort:SERVer:TTOut on page 258

Update Server Connection - Import
Updates the TCP/IP parameter and tries to establish a connection to the server.

SCPI command:
[:SOURce<hw>]:BB:IMPort:SERVer:UPDate on page 258

Server ID - Import
Indicates the connected import server. If no import server is connected, "not connected" is displayed. Seraching for a connected server is triggered with "Update Server Connection".

SCPI command:
`[:SOURce<hw>]:BB:IMPort:SERVer:ID` on page 257

Filter/Clipping - Import
Calls the menu for setting baseband filtering and clipping. The current filter and the clipping state are displayed next to the button.

The menu is described in chapter 5.10.6.3, "Filter / Clipping Settings", on page 118.

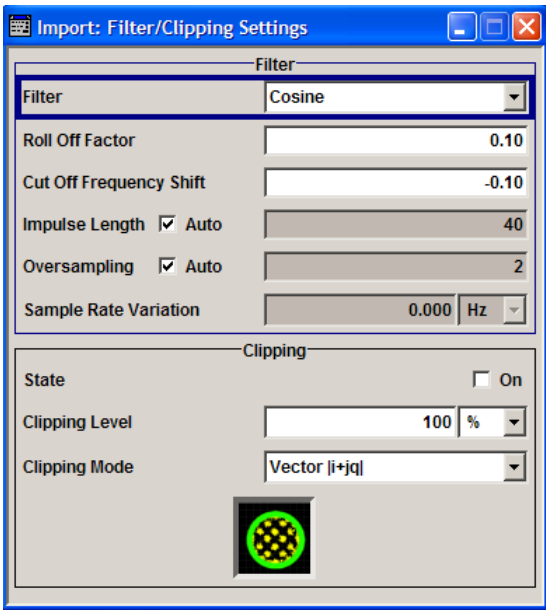
Marker - Import
Calls the menu for configuring the marker signals.

This menu is described in chapter 5.10.6.4, "Marker Menu", on page 121.

5.10.6.3 Filter / Clipping Settings

To access this dialog, select "Import Main Menu > Filter/Clipping".

The baseband filter, sample rate variation and clipping are defined in this menu.



Import: Filter/Clipping Settings

Filter

Filter	Cosine
Roll Off Factor	0.10
Cut Off Frequency Shift	-0.10
Impulse Length ☑ Auto	40
Oversampling ☑ Auto	2
Sample Rate Variation	0.000 Hz

Clipping

State	□ On
Clipping Level	100 %
Clipping Mode	Vector |i+jq|

Filter Settings

Filter
Sets the baseband filter.

SCPI command:
`[:SOURce<hw>]:BB:IMPort:FILTer:TYPE` on page 262

Roll Off Factor or BxT

Sets the filter parameter.

For the default cosine filter, a roll off factor of 0.10 is used.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:COSine` on page 262
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:RCOSine` on page 263
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:PGAuss` on page 263
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:GAUSs` on page 262
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:SPHase` on page 263
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:APCO25` on page 263

Cut Off Frequency Shift

(available for filter parameter Cosine only)

Sets the value for the cut off frequency shift. The cut off frequency of the cosine filter can be adjusted to reach spectrum mask requirements.

The value range is -1.0 to 1.0.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:COSine:COFS` on page 263

Cut Off Frequency Factor

(available for filter parameter Lowpass and EUTRA/LTE only)

Sets the value for the cut off frequency factor. The cut off frequency of the filter can be adjusted to reach spectrum mask requirements.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:LPASs` on page 262
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:LPASSEVM` on page 263

Sample Rate Variation

Sets the sample rate of the signal. A variation of this parameter only affects the ARB clock rate; all other signal parameters remain unchanged.

If the sampling rate in the "General Settings" menu is changed, this parameter is reset to the chosen sampling rate.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:SRATe:VARiation` on page 259

Impulse Length

Displays the number of filter tabs. If the check box is activated, the most sensible parameter values are selected. The value depends on the coherence check. If the check box is deactivated, the values can be changed manually.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:FILTer:ILENgth:AUTO` on page 261
`[:SOURce<hw>]:BB:IMPort:FILTer:ILENgth` on page 261

Oversampling

Determines the upsampling factor. If the check box is activated, the most sensible parameter values are selected. The value depends on the coherence check. If the check box is deactivated, the values can be changed manually.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:FILTer:OSAMpling:AUTO` on page 262

`[:SOURce<hw>]:BB:IMPort:FILTer:OSAMpling` on page 262

Clipping Settings

The settings for clipping are collected in the "Clipping" section.

Clipping State

Switches baseband clipping on and off.

Baseband clipping is a very simple and effective way of reducing the crest factor of the signal.

With baseband clipping, the signal level is limited to a settable value ("Clipping Level"). This level is specified as a percentage of the highest peak value. Since clipping is done prior to filtering, the procedure does not influence the spectrum. The EVM however increases.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:CLIPping:STATe` on page 261

Clipping Level

Sets the limit for clipping.

This value indicates at what point the signal is clipped. It is specified as a percentage, relative to the highest level. 100% indicates that clipping does not take place.

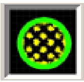
SCPI command:

`[:SOURce<hw>]:BB:IMPort:CLIPping:LEVel` on page 260

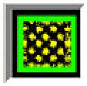
Clipping Mode

Selects the clipping method. A graphic illustration of the way in which these two methods work is given in the menu.

"Vector | i + q |" The limit is related to the amplitude | i + q |. The I and Q components are mapped together, the angle is retained.



"Scalar | i | + | q |" The limit is related to the absolute maximum of all the I and Q values | i | + | q | .



The I and Q components are mapped separately, the angle changes.

SCPI command:

`[:SOURce<hw>]:BB:IMPort:CLIPping:MODE` on page 261

5.10.6.4 Marker Menu

The marker output signal for synchronizing external instruments is configured in the "Marker Settings" section "Marker Mode".

Marker Mode

Selects a marker signal for the associated "MARKER" output.

"Restart" A marker signal is generated at the start of each ARB sequence.

"Pulse" A regular marker signal is generated. The pulse frequency is defined by entering a divider. The frequency is derived by dividing the sample rate by the divider. The input box for the divider opens when "Pulse" is selected, and the resulting pulse frequency is displayed below it.

SCPI command:

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PULSe:DIVider on page 265

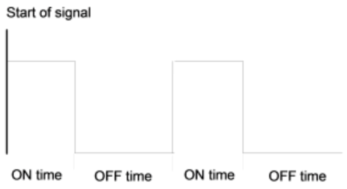
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PULSe:FREQuency on page 265

"Pattern " A marker signal that is defined by a bit pattern is generated. The pattern has a maximum length of 64 bits and is defined in an input field which opens when "Pattern" is selected.

SCPI command:

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PATTern on page 265

"ON/OFF Period" A regular marker signal that is defined by an ON/OFF ratio is generated. A period lasts one ON and OFF cycle.



The "ON Time" and "OFF Time" are each expressed as a number of samples and are set in an input field which opens when "ON/OFF ratio" is selected.

SCPI command:

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:OFFTime on page 264

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:ONTime on page 264

SCPI command:

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:MODE on page 264

5.11 AWGN - Noise Generator

The noise generator generates an AWGN signal (Additive White Gaussian Noise) in the digital baseband, i.e. the noise power density has a Gaussian distribution and is distributed evenly across the frequency. This noise signal superimposes the (interference-free) useful signal ("Additive Noise" mode). Typical applications for the noise generator are bit-error or block-error measurements, depending on the set S/N or C/N ratio.

The Gaussian noise is generated by means of feedback shift registers with subsequent probability transformation. The switching configuration is such that virtually ideal statistical characteristics are achieved:

- I and Q paths are decorrelated from each other.
- The crest factor of 18 dB allows very small probabilities to be realized.

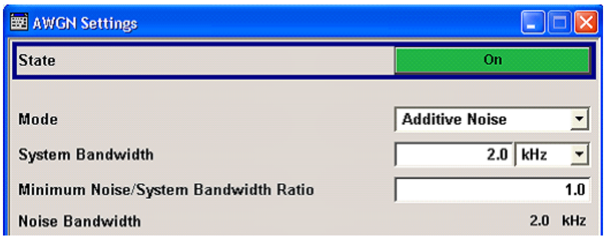
Apart from the "Additive Noise" mode, there are also the "Noise Only" and the "CW Interferer" mode. In the "Noise Only" mode a pure noise signal is generated and modulated to the carrier. In the "CW Interferer mode", a sinusoidal signal with an adjustable frequency offset to the baseband signal is generated and added to the baseband signal by means of a counter instead of a shift register.

The achievable noise bandwidth can not exceed the sampling rate of the baseband. This rate is assumed to be 50 MHz in case of Noise Only where 100.000 samples are generated.

5.11.1 AWGN Settings Dialog

The "AWGN Settings" menu is divided into the following sections.

The RFI signal generator is activated, the mode selected and the interfering signal is configured in the top section of the menu.



State: On

Mode: Additive Noise

System Bandwidth: 2.0 kHz

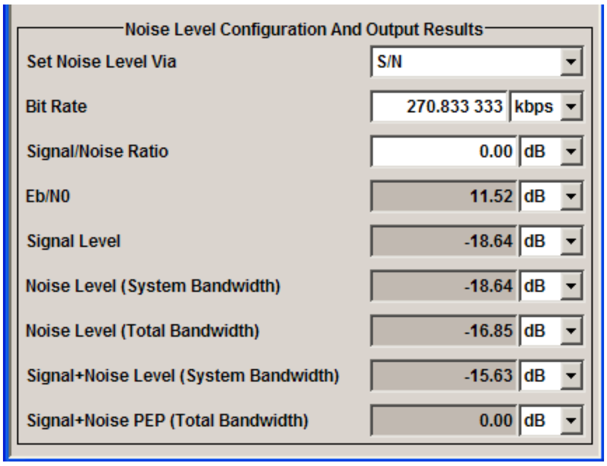
Minimum Noise/System Bandwidth Ratio: 1.0

Noise Bandwidth: 2.0 kHz

The level of the interfering signal is configured in the "Noise Level Configuration And Output Results" section.

In "Additive Noise" and "CW interferer" mode, the interfering level can be defined and the resulting total level displayed here.

In "Noise Only" mode, only the noise level can be set here.



Noise Level Configuration And Output Results

Set Noise Level Via	S/N	
Bit Rate	270.833 333	kbps
Signal/Noise Ratio	0.00	dB
Eb/N0	11.52	dB
Signal Level	-18.64	dB
Noise Level (System Bandwidth)	-18.64	dB
Noise Level (Total Bandwidth)	-16.85	dB
Signal+Noise Level (System Bandwidth)	-15.63	dB
Signal+Noise PEP (Total Bandwidth)	0.00	dB

5.11.1.1 AWGN Settings

The RFI signal generator is activated, the mode selected and the interfering signal is configured in the top section of the menu.

State - AWGN
Activates/deactivates the RFI signal generator. The interferer (AWGN or CW interferer, depending on the selected mode) is generated as after the generator is activated.

SCPI command:
`[:SOURce<hw>]:AWGN:STATe` on page 207

Mode - AWGN
Selects the mode for generating the interfering signal.

"Additive Noise" The AWGN noise signal with selectable system bandwidth is added to the baseband signal.

"Noise Only" The pure AWGN noise signal with selectable system bandwidth is modulated to the carrier. The connection to the baseband is interrupted.

"CW Interferer" A sine with a defined frequency offset is added to the baseband signal. The calculation of E_b/N_0 ratio is omitted.

SCPI command:
`[:SOURce<hw>]:AWGN:MODE` on page 203

Sample Rate
(Noise Only)

Sets the sample rate of the AWGN noise signal.

The sample rate depends on the configured instrument. If the configured instrument is changed, the value range of this parameter is changed as well.

The maximum value of the sample rate has to be lower or equal to the maximum clock frequency of the selected instrument.

SCPI command:
`[:SOURce<hw>]:AWGN:SRATe` on page 206

Sequence Length

(Noise Only)

Sets the sequence length of the signal in samples.

The sequence length depends on the configured instrument. If the configured instrument is changed, the value range of this parameter is changed as well.

SCPI command:
`[:SOURce<hw>]:AWGN:SLENgth` on page 206

System Bandwidth - AWGN

(Additive Noise and Noise Only)

Sets the RF bandwidth to which the set carrier/noise ratio relates.

Within this frequency range, the signal is superimposed with a noise signal whose level corresponds exactly to the set C/N or S/N ratio.

Note: The noise signal is not generated arbitrarily for a particular bandwidth, but instead it is generated in steps. Noise therefore also occurs outside the set system bandwidth.

SCPI command:
`[:SOURce<hw>]:AWGN:BWIDth` on page 201

Minimum Noise/System Bandwidth Ratio - AWGN

(Additive Noise and Noise Only)

Sets the ratio of minimum noise bandwidth to system bandwidth.

Note: The noise signal is not generated arbitrarily for a particular bandwidth, but instead it is generated in steps. Noise therefore also occurs outside the set system bandwidth.

With this parameter the minimum real noise bandwidth can be set as required by some standards. It does not influence the calculation of level from the set C/N or S/N ratio in relation to system bandwidth.

The value range of this parameter is automatically adjusted depending on the selected instrument. The maximum ratio minimum noise bandwidth to system bandwidth is determinate by the max sample clock of the selected instrument and is calculated so that the overall bandwidth, i.e. "System BW" x "Minimum Noise"/"System BW Ratio" has to be less or equal to the sampling rate of the selected instrument.

SCPI command:
`[:SOURce<hw>]:AWGN:BWIDth:RATio` on page 201

Noise Bandwidth - AWGN

(Additive Noise and Noise Only)

Indicates the real noise bandwidth. The value is only indicated for "State On".

Note: The noise signal is not generated arbitrarily for a particular bandwidth, but instead it is generated in steps. Noise therefore also occurs outside the set system bandwidth. The noise bandwidth can not exceed the sampling rate of the baseband signal since white Gaussian noise is generated and added offline to the baseband signal. In case of "Noise Only" mode, the noise bandwidth is also determined by and can not exceed the sampling rate set with the parameter "Sample Rate".

SCPI command:
`[:SOURce<hw>]:AWGN:BWIDth:NOISe` on page 201

Target CW Frequency - AWGN
(CW Interferer mode only)

Sets the desired frequency of the sine in "CW Interferer" mode.

The range of this parameter is limited to ± ("Symbol Rate" + "Sample Rate")/4

SCPI command:
`[:SOURce<hw>]:AWGN:FREQuency:TARGet` on page 203

Resulting CW Frequency - AWGN
(CW Interferer mode only)

Indication of the actual CW frequency of the sinusoidal signal in "CW Interferer" mode.

The resulting CW Frequency Offset is the correction of the desired value based on generating an integer multiple of periods out of the output sequence length.

SCPI command:
`[:SOURce<hw>]:AWGN:FREQuency:RESult` on page 203

5.11.1.2 Noise/Interferer Level Configuration And Output Results

The interfering signal is configured in the "Noise/Interferer Level Configuration and Output Results" section.

Set Noise Level via - AWGN
(Additive Noise only)

Selects the mode for setting the noise level

"C/N ""S/N" The noise level is set on the basis of the value entered for the carrier/noise or signal/noise ratio.
With digital modulation, the associated E_b/N_0 value is determined and displayed.

"E_b/N_0" The noise level is set on the basis of the value entered for the ratio of bit energy to noise power density.
The associated C/N or S/N value is displayed. The correlation between the two values is as follows:
C/N or S/N = $(E_b/N_0)*(f_{bit}/B_{sys})$, where
f_{bit} is the Bit rate (Symbol rate x Modulation value)
B_{sys} is the System bandwidth
With "Custom Dig Mod" signals, the bit rate used for calculation is defined by the selected standard in the "Custom Digital Mod" dialog.
With "Digital Standard" signals, the bit rate to be used for calculating the E_b/N_0 can be entered at "Bit Rate".
Some test cases with the 3GPP base station tests (TS 25.141) specify, for example, E/N settings that apply to channel-coded data or block segments.

SCPI command:
`[:SOURce<hw>]:AWGN:POWer:MODE` on page 204

Bit Rate - AWGN
(Additive Noise only)

Sets or indicates the bit rate used for converting C/N or S/N to E_b/N_0.

With "Custom Digital Mod" signals, the bit rate used is indicated here. It is defined by the choice of the standard in the "Custom Digital Mod" menu.

With "Digital Standard" signals, the bit rate used for converting C/N or S/N to E_b/N_0 is set here.

When generating a "Digital Standard", it is thus possible to select which bit rate is to be used for calculating the ratio of bit energy to noise power density, e.g. the bit rate before or after channel coding.

SCPI command:
`[:SOURce<hw>]:AWGN:BRATe` on page 200

Carrier/Noise Ratio / Signal/Noise Ratio
(Additive Noise and CW Interferer only)

Sets the carrier/noise, signal/noise or signal/inerferer ratio.

If "Set Noise Level via E_b/N_0" is selected, the associated C/N or S/N value is displayed here.

SCPI command:
`[:SOURce<hw>]:AWGN:CNRatio` on page 202

E_b/N_0 - AWGN
(Additive Noise only)

Sets the ratio of bit energy to noise power density in "Additive Noise" mode.

- The level of the noise signal which is derived from the entered E_b/N_0 value is displayed under "Noise Level (System Bandwidth)".
- The level of the useful signal is displayed in the menu under "Carrier Level".
- The level of the output signal is displayed under "Carrier Level + Noise".
- These levels are additionally scaled after the generation of the interferer/noise so that the peak of the output signal corresponds to 0 dBFS (Full Scale).

If "Set Noise Level via C/N or S/N" is selected, the associated E_b/N_0 value is displayed here.

SCPI command:
`[:SOURce<hw>]:AWGN:ENRatio` on page 202

Carrier Level / Signal Level
(Additive Noise and CW Interferer)

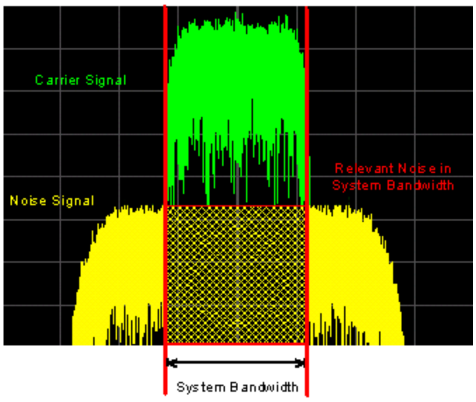
Displays the power level of the useful signal.

SCPI command:
`[:SOURce<hw>]:AWGN:POWer:CARRier` on page 204

Noise Level (System Bandwidth) / Interferer Level
(Additive Noise, Noise Only, CW Interferer)

Displays the level of the noise signal in the system bandwidth.

Note: The noise signal is not generated arbitrarily for a particular bandwidth, but instead it is generated in steps. Noise therefore also occurs outside the set system bandwidth. This means that the total measurable noise level (see below) usually exceeds the value displayed here. Correct measurement of the noise level within the system bandwidth is possible by means of channel power measurement using a signal analyzer.



SCPI command:
[:SOURce<hw>]:AWGN:POWer:NOISe on page 205

Noise Level (Total Bandwidth) - AWGN
Displays the level of the noise signal in the total bandwidth in "Noise Only" and "Additive Noise" mode.

SCPI command:
[:SOURce<hw>]:AWGN:POWer:NOISe:TOTal on page 205

Carrier + Noise Level / Signal + Noise Level (System Bandwidth) / Carrier + Interferer Level / Signal + Interferer Level
(Additive Noise and CW Interferer)

Displays the overall level of the noise/interferer signal plus useful signal.

SCPI command:
[:SOURce<hw>]:AWGN:POWer:SUM on page 205

Carrier + Noise PEP / Signal + Noise PEP (Total Bandwidth) / Carrier + Interferer PEP / Signal + Interferer PEP
(Additive Noise and CW Interferer)

Displays the peak envelope power of the overall signal comprised of noise signal plus useful signal.

This value is set to 0 dBFS where all other display power values are shown relative to the PEP after noise generation.

SCPI command:
[:SOURce<hw>]:AWGN:POWer:SUM:PEP on page 206

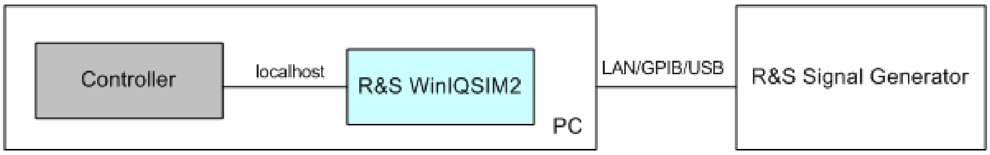
6 Automation of R&S WinIQSIM2

In addition to being able to control the R&S WinIQSIM2 in the usual manual way via the user interface, this system can also be controlled remotely. Remote control operation allows automation of the configuration process and is especially useful when a higher configuration speed is required.

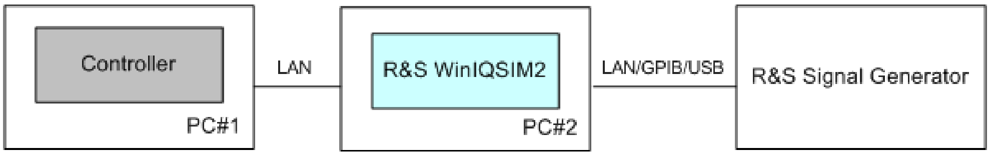
Remote control is an operation by which the software is operated remotely from a so called controller. A controller can be any PC on which some required software is installed and configured. R&S WinIQSIM2 might be but not necessarily is installed on the same PC. In case R&S WinIQSIM2 is not physically located on the controller's PC, a LAN connection between both PCs is required.

The figures below depict the following two possible cases:

- The controller and R&S WinIQSIM2 are on the same PC.



- In some rare cases, the controller and the software are installed on two separate PCs.



Remote control operation of the R&S WinIQSIM2 should not be mistaken with the functions provided in the "Remote Control (SCPI)" dialog. This dialog enables a remote control of an instrument from the graphical user interface of the software, i.e. in the general case the R&S WinIQSIM2 itself is manually controlled while the instrument is remote controlled.

In remote control operation, the software itself is operated by means of remote control commands. A detailed description of specific SCPI commands available for the R&S WinIQSIM2 is provided in chapter 7, "Remote Control Commands", on page 165. The remote control commands related to the digital standards are described in detail after each related function section.

For background information about the SCPI command structure and basic information on operating the R&S WinIQSIM2 software via remote control, refer to chapter 6.1, "Remote Control Basics", on page 129.

Refer to chapter 6.1.2, "Starting a Remote Control Session", on page 134 for information on how to get started and how to establish a remote control connection between the controller and the software.

6.1 Remote Control Basics

This chapter provides basic information on operating an instrument via remote control.

Through this description, the term “instrument” is used to indicate the R&S WinIQSIM2; the terms “instrument”, “software” and R&S WinIQSIM2 are used interchangeably.

6.1.1 Remote Control Interfaces and Protocols

The instrument supports different interfaces for remote control. The following table gives an overview.

Table 6-1: Remote control interfaces and protocols

Interface	Protocols, VISA*) address string	Remarks

Local Area Network (LAN)	Protocols:
	- VXI-11
	VISA*) address string:
	`TCPIP::host address[::LAN device name][::INSTR]`
	- socket communication (Raw Ethernet, simple telnet)
	VISA*) address string:
	`TCPIP::host address[::LAN device name]::<port>::SOCKET`
		A LAN connector is located on the front or rear panel of the instrument, or both.
		The interface is based on TCP/IP and supports various protocols.
		For a description of the protocols refer to:
		- "VXI-11 Protocol" on page 133
		- "Socket Communication" on page 133

*) VISA is a standardized software interface library providing input and output functions to communicate with instruments. A VISA installation on the controller is a prerequisite for remote control over LAN (when using VXI-11 or HiSLIP protocol), USB and serial interface. However, no VISA installation is necessary to remote control while using socket communication. For more information, see chapter 6.1.1.1, "VISA Libraries", on page 130.		

Within this interface description, the term GPIB is used as a synonym for the IEC/IEEE bus interface.

SCPI (Standard Commands for Programmable Instruments)

SCPI commands - messages - are used for remote control. Commands that are not taken from the SCPI standard follow the SCPI syntax rules. The instrument supports the SCPI version 1999. The SCPI standard is based on standard IEEE 488.2 and aims at the standardization of device-specific commands, error handling and the status registers. The tutorial "Automatic Measurement Control - A tutorial on SCPI and IEEE 488.2" from John M. Pieper (R&S order number 0002.3536.00) offers detailed information on concepts and definitions of SCPI.

Tables provide a fast overview of the bit assignment in the status registers. The tables are supplemented by a comprehensive description of the status registers.

6.1.1.1 VISA Libraries

VISA is a standardized software interface library providing input and output functions to communicate with instruments. The I/O channel (LAN or TCP/IP, USB, GPIB,...) is selected at initialization time by means of the channel–specific address string ("VISA resource string"), or by an appropriately defined VISA alias (short name). A VISA installation is a prerequisite for remote control using the following interfaces:

- "VXI-11 Protocol" on page 133

For more information about VISA refer to the user documentation.

6.1.1.2 Messages

The messages transferred on the data lines are divided into the following categories:

- Interface messages
  Interface messages are transmitted to the instrument on the data lines, with the attention line being active (LOW). They are used to communicate between the controller and the instrument. Interface messages can only be sent by instruments that have GPIB bus functionality. For details see the sections for the required interface.
- Instrument messages
  Instrument messages are employed in the same way for all interfaces, if not indicated otherwise in the description. Structure and syntax of the instrument messages are described in chapter 6.1.3, "SCPI Command Structure", on page 144. A detailed description of all messages available for the instrument is provided in the chapter "Remote Control Commands".
  There are different types of instrument messages, depending on the direction they are sent:
  - Commands
  - Instrument responses

Commands

Commands (program messages) are messages the controller sends to the instrument. They operate the instrument functions and request information. The commands are subdivided according to two criteria:

- According to the effect they have on the instrument:
  - **Setting commands** cause instrument settings such as a reset of the instrument or setting the frequency.
  - **Queries** cause data to be provided for remote control, e.g. for identification of the instrument or polling a parameter value. Queries are formed by directly appending a question mark to the command header.
- According to their definition in standards:
  - **Common commands**: their function and syntax are precisely defined in standard IEEE 488.2. They are employed identically on all instruments (if implemented). They refer to functions such as management of the standardized status registers, reset and self test.
  - **Instrument control commands** refer to functions depending on the features of the instrument such as frequency settings. Many of these commands have also been standardized by the SCPI committee. These commands are marked as "SCPI compliant" in the command reference chapters. Commands without this SCPI label are device-specific, however, their syntax follows SCPI rules as permitted by the standard.

Instrument responses

Instrument responses (response messages and service requests) are messages the instrument sends to the controller after a query. They can contain measurement results, instrument settings and information on the instrument status.

6.1.1.3 LAN Interface

For remote control via a network, the controller and the PC on that the R&S WinIQSIM2 is installed (if not the same) must be connected via the LAN interface to a common network with TCP/IP network protocol. They are connected using a commercial RJ45 cable. Software for remote control and the VISA program library must be installed on the controller.

VISA library

Instrument access via VXI-11 protocols is usually achieved from high level programming platforms using VISA as an intermediate abstraction layer. VISA encapsulates the low level VXI or GPIB function calls and thus makes the transport interface transparent for the user. See chapter 6.1.1.1, "VISA Libraries", on page 130 for details.

IP address

Only the IP address or the computer name (LAN device name) is required to set up the connection. The IP address/computer name is part of the "visa resource string" used by the programs to identify and control the instrument.

Forms of the VISA resource string:

```
TCPIP::host address[::LAN device name][::INSTR]
```

or

```
TCPIP::host address::port::SOCKET
```

where

- **TCPIP** designates the network protocol used
- **host address** is the IP address or host name of the device
- **LAN device name** defines the protocol and the instance number of a sub-instrument:
  - `inst0` selects the VXI-11 protocol (optional, default)
- **INSTR** indicates the instrument resource class (optional)
- **port** determines the used port number
- **SOCKET** indicates the raw network socket resource class

Example:

In case the R&S WinIQSIM2 and the controller are installed on the same PC, the IP address *127.0.0.1* is used. Hence the valid visa resource string is:

```
TCPIP::127.0.0.1::INSTR
```

Identifying instruments in a network

If several instruments are connected to the network, each instrument has its own IP address and associated resource string. The controller identifies these instruments by means of the resource string.

VXI-11 Protocol

The VXI-11 standard is based on the ONC RPC (Open Network Computing Remote Procedure Call) protocol which in turn relies on TCP/IP as the network/transport layer. The TCP/IP network protocol and the associated network services are preconfigured. TCP/IP ensures connection-oriented communication, where the order of the exchanged messages is adhered to and interrupted links are identified. With this protocol, messages cannot be lost.

Socket Communication

An alternative way for remote control of the software is to establish a simple network communication using sockets. The socket communication, also referred as “Raw Ethernet communication”, does not necessary require a VISA installation on the remote controller side.

The simplest way to establish socket communication is to use the built-in telnet program. The telnet program is part of every operating system and supports a communication with the software on a command-by-command basis. For better utilization and to enable automation by means of programs, user defined sockets can be programmed.

Socket connections are established on a specially defined port. The socket address is a combination of the IP address or the host name of the instrument and the number of the port configured for remote-control. All R&S Signal Generator use port number 5025 for this purpose. The port is configured for communication on a command-to-command basis and for remote control from a program.

LAN Interface Messages

In the LAN connection, the interface messages are called low–level control messages. These messages can be used to emulate interface messages of the GPIB bus.

Command	Long term	Effect on the instrument
&ABO	Abort	Aborts processing of the commands just received.
&DCL	Device Clear	Aborts processing of the commands just received and sets the command processing software to a defined initial state. Does not change the instrument setting.
&GTL	Go to Local	Transition to the "local" state (manual control).
&GTR	Go to Remote	Transition to the "remote" state (remote control).
&GET	Group Execute Trigger	Triggers a previously active instrument function (e.g. a sweep). The effect of the command is the same as with that of a pulse at the external trigger signal input.
&LLO	Local Lockout	Disables switchover from remote control to manual control by means of the front panel keys.

Command	Long term	Effect on the instrument
&NREN	Not Remote Enable	Enables switchover from remote control to manual operation by means of the front panel keys
&POL	Serial Poll	Starts a serial poll.

6.1.2 Starting a Remote Control Session

The instrument and the controller have to be connected with the suitable cable and switched on.

A remote control program must open a connection to the instrument (using VISA functionality), before it can send commands to and receive device responses from the instrument.

Refer to chapter 6.1.2.2, "Examples", on page 134 for practical examples on setting up of a remote control link and starting of a remote control session.

6.1.2.1 Switching to Remote Control

When it is started, the software is always in the manual operating state and can be operated via the block diagram.

Starting remote control

► Send a command from a controller to the signal generation software.

The software is in a remote control state as soon as it receives a command from the controller.

6.1.2.2 Examples

This sections provides examples for setting up of remote control connection and starting a remote control session over LAN and GPIB interfaces.

This section assumes basic knowledge of programming and operation of the controller. A description of the interface commands can be obtained from the relevant manuals.

Remote Control over LAN using VXI-11 Protocol

Through the examples in this section, the program 'Measurement & Automation Explorer' from National Instruments under Windows operating system is used for setting up a LAN remote control link and starting a remote control session.

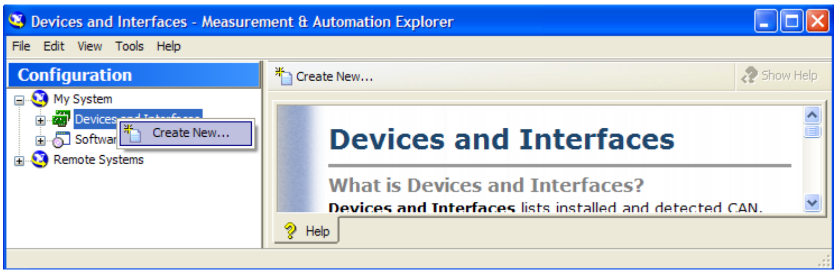
Administrator rights are necessary for installation and starting of the R&S WinIQSIM2.

For remote control (VXI11) it is recommended to allow R&S WinIQSIM2 to be accessed through port 111. Adjust the firewall settings accordingly.

Configuring the controller

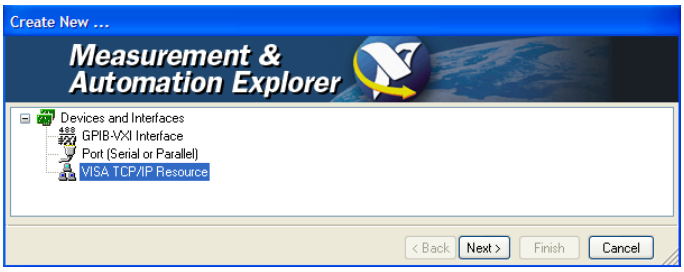
To enable the external controller to communicate with the software via TCP/IP protocol, set up a remote control link as follow:

1. Connect the controller and the instrument to the network (network cable) and switch them on.
2. Start the 'Measurement & Automation Control' program on the controller.
3. Select "Devices and Interfaces > Create New".

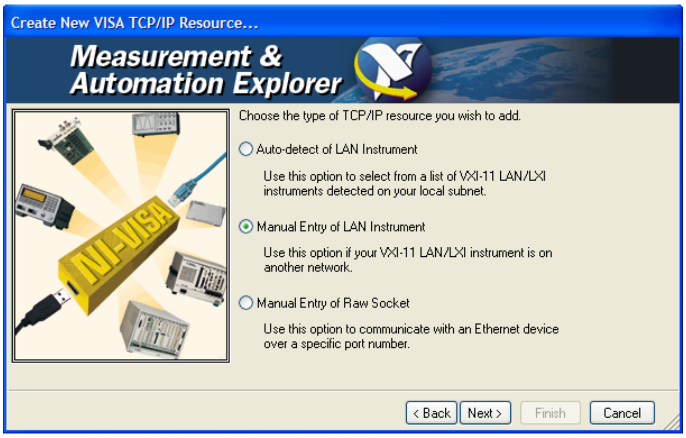


Devices and Interfaces lists installed and detected CAN.

4. Select "VISA TCP/IP Resource" and confirm with "Next".



5. Choose the type of TCP/IP resource you wish to add and select Next.


Create New VISA TCP/IP Resource...
Measurement & Automation Explorer
NI-VISA
< Back
Next >
Finish
Cancel

Choose the type of TCP/IP resource you wish to add.

○ Auto-detect of LAN Instrument

Use this option to select from a list of VXI-11 LAN/LXI instruments detected on your local subnet.

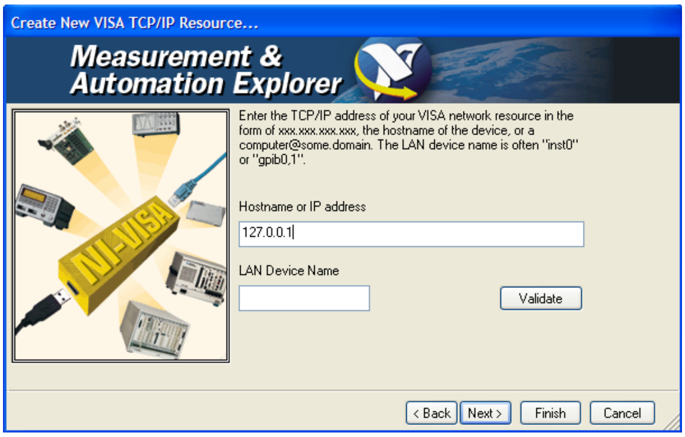
● Manual Entry of LAN Instrument

Use this option if your VXI-11 LAN/LXI instrument is on another network.

○ Manual Entry of Raw Socket

Use this option to communicate with an Ethernet device over a specific port number.

6. Enter the IP address or the host name of the R&S WinIQSIM2 and select "Next".


Create New VISA TCP/IP Resource...
Measurement & Automation Explorer
NI-VISA
< Back
Next >
Finish
Cancel
Validate

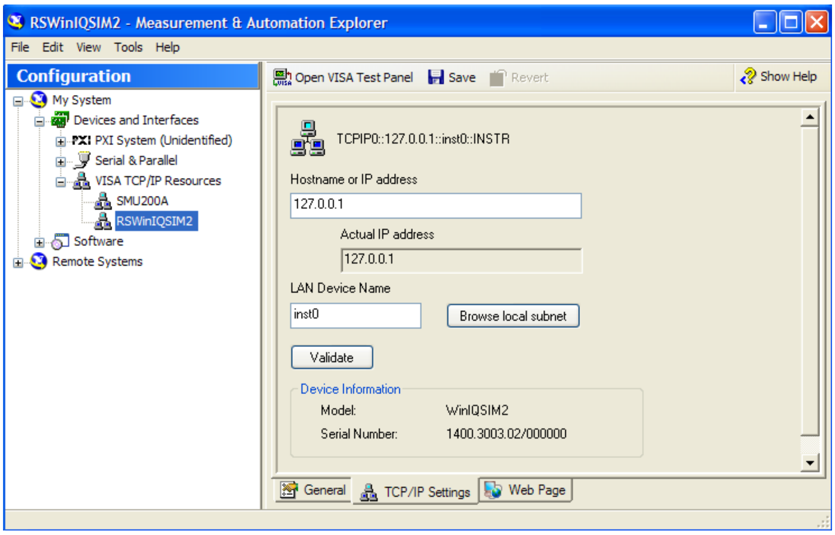
Enter the TCP/IP address of your VISA network resource in the form of xxx.xxx.xxx.xxx, the hostname of the device, or a computer@some.domain. The LAN device name is often "inst0" or "gpib0,1".

Hostname or IP address: 127.0.0.1

LAN Device Name:

7. Enter the alias name if required.
   The alias name must not be mistaken for the computer name. It is only used for instrument identification within the program and displayed in the menu as an option in case of an Ethernet link.

8. Confirm the settings with "Finish".
   The instrument is configured and the settings are displayed in the "TCP/IP Settings" tab.



TCPIP0::127.0.0.1::inst0::INSTR

Hostname or IP address: 127.0.0.1

Actual IP address: 127.0.0.1

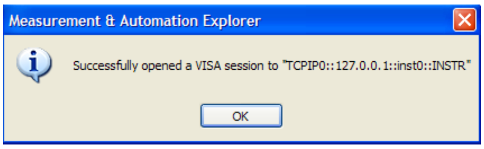
LAN Device Name: inst0

Device Information

Model: WinIQSIM2

Serial Number: 1400.3003.02/000000

9. To test the connection, select "Validate".
   A message indicates whether the link to the instrument can be set up or not.



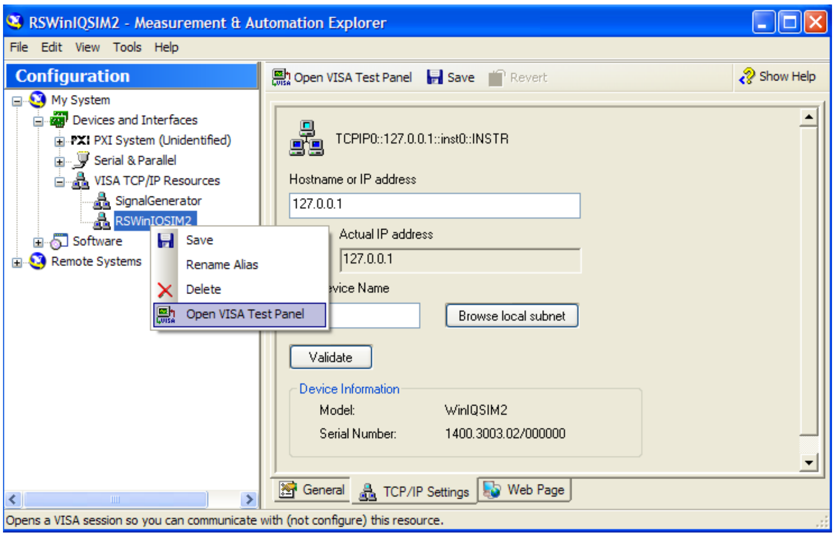
Successfully opened a VISA session to "TCPIP0::127.0.0.1::inst0::INSTR"

If a connection cannot be set up, check whether the controller and the instrument are connected to the network (network cable) and switched on. Correct spelling of the IP address or the computer name can also be checked. For further error location, inform the network administrator. In large networks, specification of additional addresses may be required for link setup, e.g. gateway and subnet mask, which are known to the network administrator.

The instrument is now registered in the program and can be addressed via the resource string or alias name.

Starting a remote control over LAN (using VXI-11)

1. Start the 'Measurement & Automation Explorer' on the controller.
2. In the "Configuration" window, select "Device and Interfaces > VISA TCP/IP Resources", select the required instrument and select "Open VISA Test Panel".



RSWinIQSIM2 - Measurement & Automation Explorer

TCPIP0::127.0.0.1::inst0::INSTR

Hostname or IP address: 127.0.0.1

Actual IP address: 127.0.0.1

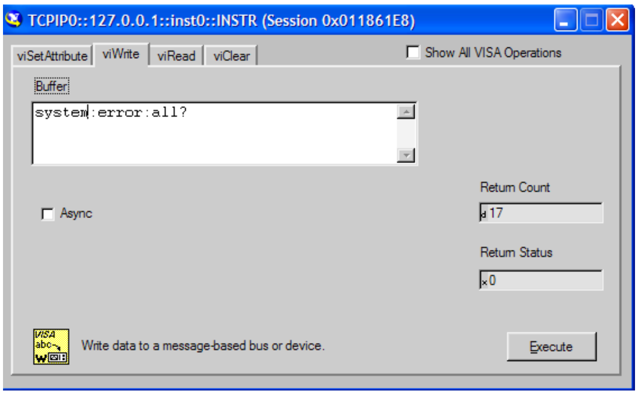
Device Information

Model: WinIQSIM2

Serial Number: 1400.3003.02/000000

Opens a VISA session so you can communicate with (not configure) this resource.

3. In the "viWrite" tab, write the command to be send to the instrument and select "Execute".



TCPIP0::127.0.0.1::inst0::INSTR (Session 0x011861E8)

Buffer: system:error:all?

□ Async

Return Count: 17

Return Status: 0

Write data to a message-based bus or device.

Instrument responses are displayed on the "viRead" tab.

Tip: For further program operation refer to the online help of the program.

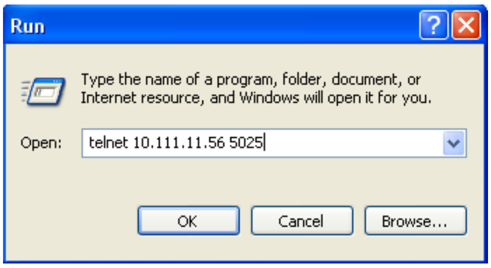
Remote Control over LAN using Socket Communication

This chapter provides an example on how to establish a remote control connection over telnet protocol and a simple sockets-based program example that can be further developed.

Setting up a Telnet Connection

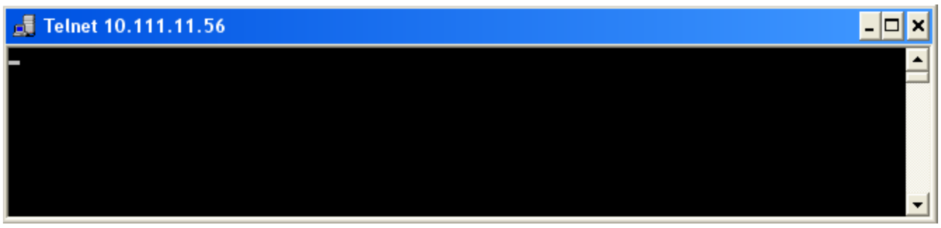
To control the software, only a telnet program is required. The telnet program is part of every operating system.

1. To establish a Telnet connection with the R&S WinIQSIM2, start the telnet program and enter the socket address.
   The socket address is a combination of the IP address or the host name of the R&S WinIQSIM2 and the number of the port configured for remote-control via telnet.
   Tip: The R&S WinIQSIM2 uses the port number 5025 for remote connection via Telnet.

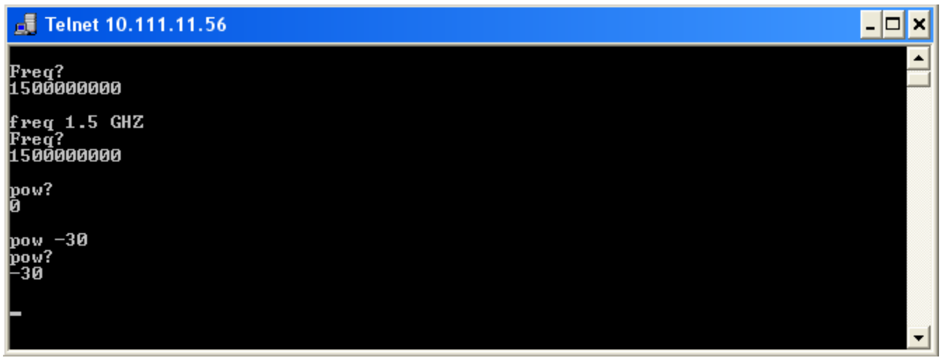
   
   Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
OK
Cancel
Browse...

   ```
   telnet 10.111.11.56 5025
   ```

   The connection to the instrument is set up and remote-control commands can be sent.

2. Even if the cursor is not visible on the screen, enter blind a remote-control command and confirm with Enter.



```
Freq?
1500000000

freq 1.5 GHZ
Freq?
1500000000

pow?
0

pow -30
pow?
-30
```

After the first remote-control command had been send, the R&S WinIQSIM2 is in a remote-controlled state.

TcpClient program examples

The following program example shows a simple TcpClient class that is intended to explain on how to get started with programming of sockets.

The example sets up a socket communication to R&S WinIQSIM2 and opens a simple user interface, very similar to the telnet, which allows input of commands. To enable real automation, further development of the program is required.

TcpClient.h

```
#include <string>
using namespace std;

//defines structs for socket handling
#ifndef LINUX
  #include "Winsock2.h"
#else
  #include <netinet/in.h>
#endif

typedef struct sockaddr_in SockAddrStruct;
typedef struct hostent HostInfoStruct;
class TcpClient
{
public:
  TcpClient();
  ~TcpClient();
  void connectToServer( string &hostname, int port );
  void disconnect( );
  void transmit( string &txString );
  void receive( string &rxString );
  string getCurrentHostName( ) const;
  int getCurrentPort( ) const;
private:
  string currentHostName;
  int currentPort;
  int currentSocketDescr;
  SockAddrStruct serverAddress;
  HostInfoStruct * currentHostInfo;
  bool clientIsConnected;
  int receiveBufferSize;
};
```

TcpClient.cpp

```
#ifdef LINUX
  #include <netdb.h>
  #include <unistd.h>
#endif

#include "TcpClient.h"
TcpClient::TcpClient()
: currentHostName( "" )
, currentPort( 0 )
, currentSocketDescr( 0 )
, serverAddress ( )
, currentHostInfo( NULL )
, clientIsConnected( false )
, receiveBufferSize( 1024 )
{}
TcpClient::~TcpClient()
{
  currentHostInfo = NULL;
}
void TcpClient::connectToServer( string &hostname, int port )
{
  currentHostInfo = gethostbyname( hostname.c_str( ) );
  if( currentHostInfo == NULL )
  {
    #ifndef LINUX
      int error = WSAGetLastError();
      switch (error)
      {
        case WSANOTINITIALISED: throw string(" A successful case WSAStartup
call must occur before using this function."); break;
        case WSAENETDOWN: throw string
(" The network subsystem has failed."); break;
        case WSAHOST_NOT_FOUND: throw string
(" Authoritative answer host not found."); break;
        case WSATRY_AGAIN: throw string
(" Nonauthoritative host not found, or server failure."); break;
        case WSANO_RECOVERY: throw string
(" A nonrecoverable error occurred."); break;
        case WSANO_DATA: throw string
(" Valid name, no data record of requested type."); break;
        case WSAEINPROGRESS: throw string(" A blocking Windows Sockets 1.1 call
is in progress, or the service provider is still processing a callback function.")
; break;
    case WSAEFAULT: throw string
(" The name parameter is not a valid part of the user address space."); break;
    case WSAEINTR: throw string
(" A blocking Windows Socket 1.1 call was canceled through case
WSACancelBlockingCall."); break;
   }
  #endif

  currentHostName = "";
  currentPort = 0;
  currentHostInfo = NULL;
  clientIsConnected = false;
  throw string("error connecting host\n" );
 }
 else
 {
  currentHostName = hostname;
  currentPort = port;
  currentSocketDescr = socket(AF_INET, SOCK_STREAM, 0);
  if( currentSocketDescr == 0 )
  {
   currentHostName = "";
   currentPort = 0;
   currentHostInfo = NULL;
   clientIsConnected = false;
   throw string("can't create socket\n" );
  }
  serverAddress.sin_family = currentHostInfo->h_addrtype;
  serverAddress.sin_port = htons( currentPort );
  memcpy( (char *) &serverAddress.sin_addr.s_addr, currentHostInfo->
h_addr_list[0], currentHostInfo->h_length );
  if( connect( currentSocketDescr, ( struct sockaddr *) &serverAddress,
sizeof(serverAddress ) ) < 0 )
  {
   throw string("can't connect server\n" );
  }
  clientIsConnected = true;
 }
}
void TcpClient::disconnect( )
{
 if( clientIsConnected )
 {
  #ifndef LINUX
   closesocket( currentSocketDescr );
  #else
   close( currentSocketDescr );
  #endif
 }
 currentSocketDescr = 0;
 currentHostName = "";
 currentPort = 0;
 currentHostInfo = NULL;
 clientIsConnected = false;
}
void TcpClient::transmit( string &txString )
{
 if( !clientIsConnected )
 {
  throw string("connection must be established before any data can be sent\n");
 }
 char * transmitBuffer = new char[txString.length() +1];
 memcpy( transmitBuffer, txString.c_str(), txString.length() );
 transmitBuffer[txString.length()] = '\n'; //newline is needed!
 if( send( currentSocketDescr, transmitBuffer, txString.length() + 1, 0 ) < 0 )
 {
  throw string("can't transmit data\n");
 }
 delete [] transmitBuffer;
}
void TcpClient::receive( string &rxString )
{
 if( !clientIsConnected )
 {
  throw string("connection must be established before any data can be received\n");
 }
 char * receiveBuffer = new char[receiveBufferSize];
 memset( receiveBuffer, 0, receiveBufferSize );
 bool receiving = true;
 while( receiving )
 {
  int receivedByteCount = recv( currentSocketDescr, receiveBuffer,
   receiveBufferSize, 0 );
  if( receivedByteCount < 0 )
  {
   throw string("error while receiving data\n");
  }
  rxString += string( receiveBuffer );
  receiving = ( receivedByteCount == receiveBufferSize );
 }
 delete [] receiveBuffer;
}
string TcpClient::getCurrentHostName( ) const
{
 return currentHostName;
}
int TcpClient::getCurrentPort( ) const
{
 return currentPort;
}
```

TelnetClient.cpp

```
//TelnetClient.cpp: Defines the entry point for the console application.//

#include <iostream>
#include "TcpClient.h"
void printUsage()
{
  cout<<"usage: EthernetRawCommand <server-ip> [scpi-command]"<<endl;
}
int main( int argc, char *argv[] )
{
  int errorCode = 0; //no error
  bool useSingleCommand = false;
  string singleCommand = "";
  string hostname = "";
  int port = 5025;
  string input = "";
  TcpClient client;
  switch( argc )
  {
  case 3:
    useSingleCommand = true;
    singleCommand = argv[2];
  case 2:
    hostname = argv[1];
    break;
  default:
    printUsage();
    getchar();
    return(-1);
  }
  try
  {
#ifndef LINUX
    WORD wVersionRequested = MAKEWORD( 2, 2 );
    WSADATA wsaData;
    int err = WSAStartup( wVersionRequested, &wsaData );
    if ( err != 0 )
    {
      throw string ("Could not find a usable WinSock DLL.");
    }
#endif
    client.connectToServer( hostname, port );
    bool terminate = false;
    while( !terminate )
    {
      char buffer[1024];
      if( useSingleCommand )
      {
        input = singleCommand; //send string
      }
      else
      {
        cout <<">";
        cin.getline( buffer, 1024 );
        input = buffer;
        if( input == "end" )
        {
          terminate = true;
        }
      }
      if( !terminate)
      {
        client.transmit( input ); //send string
        int qPos = input.find( "?", 0 );
        //receive string only when needed
        if( qPos > 0 )
        {
          string rcStr = "";
          client.receive( rcStr );
          cout << rcStr;
        }
      }
      if( useSingleCommand )
      {
        terminate = true;
      }
    }
  }catch( const string errorString )
  {
    cout<<errorString<<endl<<endl;
    cout<<"Press Enter to exit"<<endl;
    getchar();
  }
  client.disconnect( );
  return errorCode;
}
```

6.1.3 SCPI Command Structure

SCPI commands consist of a so-called header and, in most cases, one or more parameters. The header and the parameters are separated by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). The headers may consist of several mnemonics (keywords). Queries are formed by appending a question mark directly to the header.

The commands can be either device-specific or device-independent (common commands). Common and device-specific commands differ in their syntax.

6.1.3.1 Syntax for Common Commands

Common (=device-independent) commands consist of a header preceded by an asterisk (*) and possibly one or more parameters.

Examples:

*RST	RESET	Resets the instrument.
*ESE	EVENT STATUS ENABLE	Sets the bits of the event status enable registers.
*ESR?	EVENT STATUS QUERY	Queries the contents of the event status register.
*IDN?	IDENTIFICATION QUERY	Queries the instrument identification string.

6.1.3.2 Syntax for Device-Specific Commands

Not all commands used in the following examples are necessarily implemented in the instrument.

For demonstration purposes only, assume the existence of the following commands for this section:

- `DISPlay[:WINDow<1...4>]:MAXimize <Boolean>`
- `FORMat:READings:DATA <type>[,<length>]`
- `HCOPy:DEVice:COLor <Boolean>`
- `HCOPy:DEVice:CMAP:COLor:RGB <red>,<green>,<blue>`
- `HCOPy[:IMMediate]`
- `HCOPy:ITEM:ALL`
- `HCOPy:ITEM:LABel <string>`
- `HCOPy:PAGE:DIMensions:QUADrant[<N>]`
- `HCOPy:PAGE:ORIentation LANDscape | PORTrait`
- `HCOPy:PAGE:SCALe <numeric value>`
- `MMEMory:COPY <file_source>,<file_destination>`
- `SENSE:BANDwidth|BWIDth[:RESolution] <numeric_value>`
- `SENSe:FREQuency:STOP <numeric value>`
- `SENSe:LIST:FREQuency <numeric_value>{,<numeric_value>}`

Long and short form

The mnemonics feature a long form and a short form. The short form is marked by upper case letters, the long form corresponds to the complete word. Either the short form or the long form can be entered; other abbreviations are not permitted.

Example:

`HCOPy:DEVice:COLor ON` is equivalent to `HCOP:DEV:COL ON`.

Case-insensitivity

Upper case and lower case notation only serves to distinguish the two forms in the manual, the instrument itself is case-insensitive.

Numeric suffixes

If a command can be applied to multiple instances of an object, e.g. specific channels or sources, the required instances can be specified by a suffix added to the command. Numeric suffixes are indicated by angular brackets (<1...4>, <n>, <i>) and are replaced by a single value in the command. Entries without a suffix are interpreted as having the suffix 1.

Example:

Definition: `HCOPy:PAGE:DIMensions:QUADrant[<N>]`

Command: `HCOP:PAGE:DIM:QUAD2`

This command refers to the quadrant 2.

Different numbering in remote control

For remote control, the suffix may differ from the number of the corresponding selection used in manual operation. SCPI prescribes that suffix counting starts with 1. Suffix 1 is the default state and used when no specific suffix is specified.

Some standards define a fixed numbering, starting with 0. If the numbering differs in manual operation and remote control, it is indicated for the corresponding command.

Optional mnemonics

Some command systems permit certain mnemonics to be inserted into the header or omitted. These mnemonics are marked by square brackets in the description. The instrument must recognize the long command to comply with the SCPI standard. Some commands are considerably shortened by these optional mnemonics.

Example:

Definition: `HCOPy[:IMMediate]`

Command: `HCOP:IMM` is equivalent to `HCOP`

Optional mnemonics with numeric suffixes

Do not omit an optional mnemonic if it includes a numeric suffix that is relevant for the effect of the command.

Example:

Definition:`DISPlay[:WINDow<1...4>]:MAXimize <Boolean>`

Command: `DISP:MAX ON` refers to window 1.

In order to refer to a window other than 1, you must include the optional `WINDow` parameter with the suffix for the required window.

`DISP:WIND2:MAX ON` refers to window 2.

Parameters

Parameters must be separated from the header by a "white space". If several parameters are specified in a command, they are separated by a comma (,). For a description of the parameter types, refer to chapter 6.1.3.3, "SCPI Parameters", on page 148.

Example:

Definition:`HCOPy:DEVice:CMAP:COLor:RGB <red>,<green>,<blue>`

Command:`HCOP:DEV:CMAP:COL:RGB 3,32,44`

Special characters

|	**Parameters**
	A vertical stroke in parameter definitions indicates alternative possibilities in the sense of "or". The effect of the command differs, depending on which parameter is used.
	Example:
	Definition:`HCOPy:PAGE:ORIentation LANDscape | PORTrait`
	Command `HCOP:PAGE:ORI LAND` specifies landscape orientation
	Command `HCOP:PAGE:ORI PORT` specifies portrait orientation
	Mnemonics
	A selection of mnemonics with an identical effect exists for several commands. These mnemonics are indicated in the same line; they are separated by a vertical stroke. Only one of these mnemonics needs to be included in the header of the command. The effect of the command is independent of which of the mnemonics is used.
	Example:
	Definition`SENSE:BANDwidth|BWIDth[:RESolution] <numeric_value>`
	The two following commands with identical meaning can be created:
	`SENS:BAND:RES 1`
	`SENS:BWID:RES 1`

[ ]	Mnemonics in square brackets are optional and may be inserted into the header or omitted. **Example**: `HCOPy[:IMMediate]` `HCOP:IMM` is equivalent to `HCOP`

{ }	Parameters in curly brackets are optional and can be inserted once or several times, or omitted.
	Example: `SENSe:LIST:FREQuency <numeric_value>{,<numeric_value>}`
	The following are valid commands:
	`SENS:LIST:FREQ 10`
	`SENS:LIST:FREQ 10,20`
	`SENS:LIST:FREQ 10,20,30,40`

6.1.3.3 SCPI Parameters

Many commands are supplemented by a parameter or a list of parameters. The parameters must be separated from the header by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). Allowed parameters are:

- Numeric values
- Special numeric values
- Boolean parameters
- Text
- Character strings
- Block data

The parameters required for each command and the allowed range of values are specified in the command description.

Numeric values

Numeric values can be entered in any form, i.e. with sign, decimal point and exponent. Values exceeding the resolution of the instrument are rounded up or down. The mantissa may comprise up to 255 characters, the exponent must lie inside the value range -32000 to 32000. The exponent is introduced by an "E" or "e". Entry of the exponent alone is not allowed. In the case of physical quantities, the unit can be entered. Allowed unit prefixes are G (giga), MA (mega), MOHM and MHZ are also allowed), K (kilo), M (milli), U (micro) and N (nano). If the unit is missing, the basic unit is used.

Example: `SENS:FREQ:STOP 1.5GHz` = `SENS:FREQ:STOP 1.5E9`

Units

For physical quantities, the unit can be entered. Allowed unit prefixes are:

- G (giga)
- MA (mega), MOHM, MHZ
- K (kilo)
- M (milli)
- U (micro)
- N (nano)

If the unit is missing, the basic unit is used.

Example:

`SENSe:FREQ:STOP 1.5GHz` = `SENSe:FREQ:STOP 1.5E9`

Some settings allow relative values to be stated in percent. According to SCPI, this unit is represented by the `PCT` string.

Example:

```
HCOP:PAGE:SCAL 90PCT
```

Special numeric values

The texts listed below are interpreted as special numeric values. In the case of a query, the numeric value is provided.

- **MIN/MAX**
  MINimum and MAXimum denote the minimum and maximum value.
- **DEF**
  DEFault denotes a preset value which has been stored in the EPROM. This value conforms to the default setting, as it is called by the *RST command.
- **UP/DOWN**
  UP, DOWN increases or reduces the numeric value by one step. The step width can be specified via an allocated step command for each parameter which can be set via UP, DOWN.
- **INF/NINF**

INFinity, Negative INFinity (NINF) represent the numeric values 9.9E37 or -9.9E37, respectively. INF and NINF are only sent as instrument responses.

- **NAN**
  Not A Number (NAN) represents the value 9.91E37. NAN is only sent as a instrument response. This value is not defined. Possible causes are the division of zero by zero, the subtraction of infinite from infinite and the representation of missing values.

Example:

Setting command: `SENSe:LIST:FREQ MAXimum`

Query: `SENS:LIST:FREQ?, Response: 3.5E9`

Queries for special numeric values

The numeric values associated to `MAXimum/MINimum/DEFault` can be queried by adding the corresponding mnemonics to the command. They must be entered following the quotation mark.

Example: `SENSe:LIST:FREQ? MAXimum`

Returns the maximum numeric value as a result.

Boolean Parameters

Boolean parameters represent two states. The "ON" state (logically true) is represented by "ON" or a numeric value 1. The "OFF" state (logically untrue) is represented by "OFF" or the numeric value 0. The numeric values are provided as the response for a query.

Example:

Setting command: `HCOPy:DEV:COL ON`

Query: `HCOPy:DEV:COL?`

Response: `1`

Text parameters

Text parameters observe the syntactic rules for mnemonics, i.e. they can be entered using a short or long form. Like any parameter, they have to be separated from the header by a white space. In the case of a query, the short form of the text is provided.

Example:

Setting command: `HCOPy:PAGE:ORIentation LANDscape`

Query: `HCOP:PAGE:ORI?`

Response: `LAND`

Character strings

Strings must always be entered in quotation marks (' or ").

Example:

`HCOP:ITEM:LABel "Test1"` or `HCOP:ITEM:LABel 'Test1'`

Block data

Block data is a format which is suitable for the transmission of large amounts of data. A command using a block data parameter has the following structure:

Example:

```
FORMat:READings:DATA  #45168xxxxxxxx
```

The ASCII character # introduces the data block. The next number indicates how many of the following digits describe the length of the data block. In the example the 4 following digits indicate the length to be 5168 bytes. The data bytes follow. During the transmission of these data bytes all end or other control signs are ignored until all bytes are transmitted.

#0 specifies a data block of indefinite length. The use of the indefinite format requires a `NL^END` message to terminate the data block. This format is useful when the length of the transmission is not known or if speed or other considerations prevent segmentation of the data into blocks of definite length.

6.1.3.4 Overview of Syntax Elements

The following table provides an overview of the syntax elements:

:	The colon separates the mnemonics of a command. In a command line the separating semicolon marks the uppermost command level.
;	The semicolon separates two commands of a command line. It does not alter the path.
,	The comma separates several parameters of a command.
?	The question mark forms a query.
*****	The asterisk marks a common command.
' ' **"**	Quotation marks introduce a string and terminate it (both single and double quotation marks are possible).
#	The hash symbol introduces binary, octal, hexadecimal and block data. ● Binary: #B10110 ● Octal: #O7612 ● Hexa: #HF3A7 ● Block: #21312
	A "white space" (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates the header from the parameters.

6.1.3.5 Structure of a command line

A command line may consist of one or several commands. It is terminated by one of the following:

- a <New Line>

- a <New Line> with EOI
- an EOI together with the last data byte

Several commands in a command line must be separated by a semicolon ";". If the next command belongs to a different command system, the semicolon is followed by a colon.

Example:

```
MMEM:COPY "Test1","MeasurementXY";:HCOP:ITEM ALL
```

This command line contains two commands. The first command belongs to the MMEM system, the second command belongs to the HCOP system.

If the successive commands belong to the same system, having one or several levels in common, the command line can be abbreviated. To this end, the second command after the semicolon starts with the level that lies below the common levels. The colon following the semicolon must be omitted in this case.

Example:

```
HCOP:ITEM ALL;:HCOP:IMM
```

This command line contains two commands. Both commands are part of the `HCOP` command system, i.e. they have one level in common.

When abbreviating the command line, the second command begins with the level below HCOP. The colon after the semicolon is omitted. The abbreviated form of the command line reads as follows:

```
HCOP:ITEM ALL;IMM
```

A new command line always begins with the complete path.

Example:

```
HCOP:ITEM ALL
HCOP:IMM
```

6.1.3.6 Responses to Queries

A query is defined for each setting command unless explicitly specified otherwise. It is formed by adding a question mark to the associated setting command. According to SCPI, the responses to queries are partly subject to stricter rules than in standard IEEE 488.2.

- The requested parameter is transmitted without a header.
  Example: `HCOP:PAGE:ORI?`, Response: `LAND`
- Maximum values, minimum values and all other quantities that are requested via a special text parameter are returned as numeric values.
  Example: `SENSe:FREQuency:STOP? MAX`, Response: `3.5E9`
- Numeric values are output without a unit. Physical quantities are referred to the basic units or to the units set using the `Unit` command. The response `3.5E9` in the previous example stands for 3.5 GHz.
- Truth values (Boolean values) are returned as `0` (for OFF) and `1` (for ON).

Example:
Setting command: `HCOPy:DEV:COL ON`
Query: `HCOPy:DEV:COL?`
Response: `1`

- Text (character data) is returned in a short form.
  Example:
  Setting command: `HCOPy:PAGE:ORIentation LANDscape`
  Query: `HCOP:PAGE:ORI?`
  Response: `LAND`

6.1.4 Command Sequence and Synchronization

IEEE 488.2 defines a distinction between overlapped and sequential commands:

- A sequential command is one which finishes executing before the next command starts executing. Commands that are processed quickly are usually implemented as sequential commands. Sequential commands are not implemented in the instrument, however the execution time of most commands is so short that they act as sequential commands when sent in different command lines.
- An overlapping command is one which does not automatically finish executing before the next command starts executing. Usually, overlapping commands take longer to process and allow the program to do other tasks while being executed. If overlapping commands do have to be executed in a defined order, e.g. in order to avoid wrong measurement results, they must be serviced sequentially. This is called synchronization between the controller and the instrument.

Setting commands within one command line, even though they may be implemented as sequential commands, are not necessarily serviced in the order in which they have been received. In order to make sure that commands are actually carried out in a certain order, each command must be sent in a separate command line.

Example: Commands and queries in one message

The response to a query combined in a program message with commands that affect the queried value is not predictable.

The following commands always return the specified result:

```
:FREQ:STAR 1GHZ;SPAN 100 :FREQ:STAR?
```

Result:

`1000000000` (1 GHz)

Whereas the result for the following commands is not specified by SCPI:

```
:FREQ:STAR 1GHz;STAR?;SPAN 1000000
```

The result could be the value of `STARt` before the command was sent since the instrument might defer executing the individual commands until a program message terminator is received. The result could also be 1 GHz if the instrument executes commands as they are received.

As a general rule, send commands and queries in different program messages.

Example: Overlapping command with *OPC

The instrument implements `INITiate[:IMMediate]` as an overlapped command. Assuming that `INITiate[:IMMediate]` takes longer to execute than `*OPC`, sending the following command sequence results in initiating a sweep and, after some time, setting the `OPC` bit in the `ESR`:

`INIT; *OPC.`

Sending the following commands still initiates a sweep:

`INIT; *OPC; *CLS`

However, since the operation is still pending when the instrument executes `*CLS`, forcing it into the "Operation Complete Command Idle" State (OCIS), `*OPC` is effectively skipped. The `OPC` bit is not set until the instrument executes another `*OPC` command.

6.1.4.1 Preventing Overlapping Execution

To prevent an overlapping execution of commands, one of the commands *OPC, *OPC? or *WAI can be used. All three commands cause a certain action only to be carried out after the hardware has been set. By suitable programming, the controller can be forced to wait for the corresponding action to occur.

*Table 6-2: Synchronization using *OPC, *OPC? and *WAI*

Command	Action	Programming the controller
*OPC	Sets the Operation Complete bit in the ESR after all previous commands have been executed.	● Setting bit 0 in the ESE ● Setting bit 5 in the SRE ● Waiting for service request (SRQ)
*OPC?	Stops command processing until 1 is returned. This is only the case after the Operation Complete bit has been set in the ESR. This bit indicates that the previous setting has been completed.	Sending *OPC? directly after the command whose processing should be terminated before other commands can be executed.
*WAI	Stops further command processing until all commands sent before *WAI have been executed.	Sending *WAI directly after the command whose processing should be terminated before other commands are executed.

Command synchronization using *WAI or *OPC? appended to an overlapped command is a good choice if the overlapped command takes only little time to process. The two synchronization techniques simply block overlapped execution of the command.

For time consuming overlapped commands it is usually desirable to allow the controller or the instrument to do other useful work while waiting for command execution. Use one of the following methods:

***OPC with a service request**

1. Set the OPC mask bit (bit no. 0) in the ESE: *ESE 1

2. Set bit no. 5 in the SRE: *SRE 32 to enable ESB service request.
3. Send the overlapped command with *OPC
4. Wait for a service request

   The service request indicates that the overlapped command has finished.

*OPC? with a service request

1. Set bit no. 4 in the SRE: *SRE 16 to enable MAV service request.
2. Send the overlapped command with *OPC?
3. Wait for a service request

   The service request indicates that the overlapped command has finished.

Event Status Register (ESE)

1. Set the OPC mask bit (bit no. 0) in the ESE: *ESE 1
2. Send the overlapped command without *OPC, *OPC? or *WAI
3. Poll the operation complete state periodically (by means of a timer) using the sequence: *OPC; *ESR?

   A return value (LSB) of 1 indicates that the overlapped command has finished.

*OPC? with short timeout

1. Send the overlapped command without *OPC, *OPC? or *WAI
2. Poll the operation complete state periodically (by means of a timer) using the sequence: <short timeout>; *OPC?
3. A return value (LSB) of 1 indicates that the overlapped command has finished. In case of a timeout, the operation is ongoing.
4. Reset timeout to former value
5. Clear the error queue with `SYStem:ERRor?` to remove the "-410, Query interrupted" entries.

Using several threads in the controller application

As an alternative, provided the programming environment of the controller application supports threads, separate threads can be used for the application GUI and for controlling the instrument(s) via SCPI.

A thread waiting for a *OPC? thus will not block the GUI or the communication with other instruments.

6.1.5 Status Reporting System

The status reporting system stores all information on the current operating state of the instrument, and on errors which have occurred. This information is stored in the status registers and in the error queue. Both can be queried with the commands `STATus...`.

6.1.5.1 Hierarchy of status registers

As shown in the following figure, the status information is of hierarchical structure.

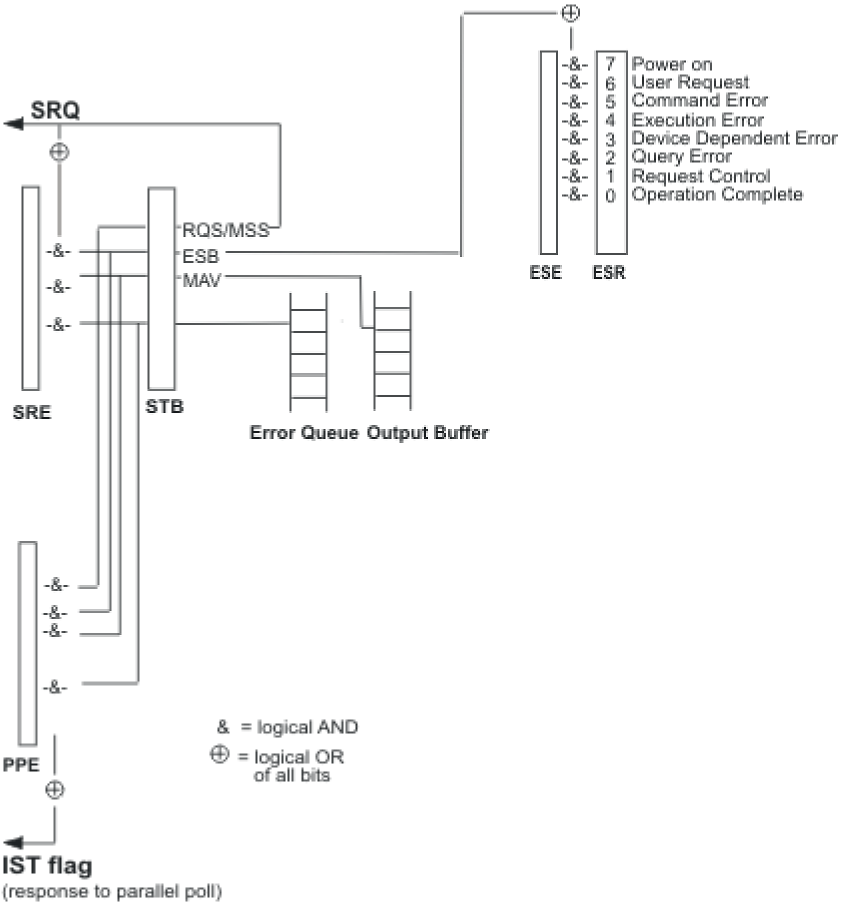


Fig. 6-1: Graphical overview of the status registers hierarchy

- **STB, SRE**
  The STatus Byte (`STB`) register and its associated mask register Service Request Enable (`SRE`) form the highest level of the status reporting system. The `STB` provides a rough overview of the instrument status, collecting the information of the lower-level registers.
- **ESR, SCPI registers**
  The STB receives its information from the following registers:
  – The Event Status Register (`ESR`) with the associated mask register standard Event Status Enable (`ESE`).

- The STATus:OPERation and STATus:QUEStionable registers which are defined by SCPI and contain detailed information on the instrument.

- **IST, PPE**
  The IST flag ("Individual STatus"), like the SRQ, combines the entire instrument status in a single bit. The PPE fulfills the same function for the IST flag as the SRE for the service request.
- **Output buffer**
  The output buffer contains the messages the instrument returns to the controller. It is not part of the status reporting system but determines the value of the MAV bit in the STB and thus is represented in the overview.

All status registers have the same internal structure.

SRE, ESE

The service request enable register SRE can be used as ENABle part of the STB if the STB is structured according to SCPI. By analogy, the ESE can be used as the ENABle part of the ESR.

6.1.5.2 Structure of a SCPI Status Register

Each standard SCPI register consists of 5 parts. Each part has a width of 16 bits and has different functions. The individual bits are independent of each other, i.e. each hardware status is assigned a bit number which is valid for all five parts. Bit 15 (the most significant bit) is set to zero for all parts. Thus the contents of the register parts can be processed by the controller as positive integers.

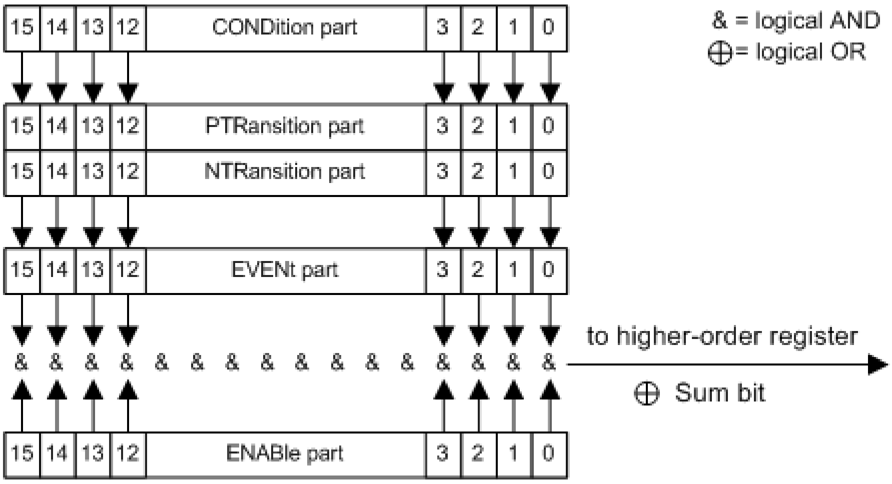

15 14 13 12 CONDition part 3 2 1 0
15 14 13 12 PTRansition part 3 2 1 0
15 14 13 12 NTRansition part 3 2 1 0
15 14 13 12 EVENt part 3 2 1 0
& & & & & & & & & & & & & & & &
15 14 13 12 ENABle part 3 2 1 0
& = logical AND
⊕= logical OR
to higher-order register
⊕ Sum bit

Fig. 6-2: The status-register model

Description of the five status register parts

The five parts of a SCPI register have different properties and functions:

- **CONDition**

The `CONDition` part is written into directly by the hardware or the sum bit of the next lower register. Its contents reflect the current instrument status. This register part can only be read, but not written into or cleared. Its contents are not affected by reading.

- **PTRansition**
  The two transition register parts define which state transition of the `CONDition` part (none, 0 to 1, 1 to 0 or both) is stored in the `EVENt` part.
  The `Positive-TRansition` part acts as a transition filter. When a bit of the `CONDition` part is changed from 0 to 1, the associated `PTR` bit decides whether the `EVENt` bit is set to 1.
  – PTR bit =1: the `EVENt` bit is set.
  – PTR bit =0: the `EVENt` bit is not set.

  This part can be written into and read as required. Its contents are not affected by reading.
- **NTRansition**
  The `Negative-TRansition` part also acts as a transition filter. When a bit of the `CONDition` part is changed from 1 to 0, the associated `NTR` bit decides whether the `EVENt` bit is set to 1.
  – NTR bit =1: the `EVENt` bit is set.
  – NTR bit =0: the `EVENt` bit is not set.

  This part can be written into and read as required. Its contents are not affected by reading.
- **EVENt**
  The `EVENt` part indicates whether an event has occurred since the last reading, it is the "memory" of the condition part. It only indicates events passed on by the transition filters. It is permanently updated by the instrument. This part can only be read by the user. Reading the register clears it. This part is often equated with the entire register.
- **ENABle**
  The `ENABle` part determines whether the associated `EVENt` bit contributes to the sum bit (see below). Each bit of the `EVENt` part is "ANDed" with the associated `ENABle` bit (symbol '&'). The results of all logical operations of this part are passed on to the sum bit via an "OR" function (symbol '+').
  `ENABle` bit = 0: the associated `EVENt` bit does not contribute to the sum bit
  `ENABle` bit = 1: if the associated `EVENt` bit is "1", the sum bit is set to "1" as well.
  This part can be written into and read by the user as required. Its contents are not affected by reading.

Sum bit

The sum bit is obtained from the `EVENt` and `ENABle` part for each register. The result is then entered into a bit of the `CONDition` part of the higher-order register.

The instrument automatically generates the sum bit for each register. Thus an event can lead to a service request throughout all levels of the hierarchy.

6.1.5.3 Contents of the Status Registers

In the following sections, the contents of the status registers are described in more detail.

Status Byte (STB) and Service Request Enable Register (SRE)

The `STatus Byte` (STB) is already defined in IEEE 488.2. It provides a rough overview of the instrument status by collecting the pieces of information of the lower registers. A special feature is that bit 6 acts as the sum bit of the remaining bits of the status byte.

The STB is read using the command `*STB` or a serial poll.

The `STatus Byte` (STB) is linked to the `Service Request Enable` (SRE) register. Each bit of the STB is assigned a bit in the SRE. Bit 6 of the SRE is ignored. If a bit is set in the SRE and the associated bit in the STB changes from 0 to 1, a service request (SRQ) is generated. The SRE can be set using the command `*SRE` and read using the command `*SRE?`.

Table 6-3: Meaning of the bits used in the status byte

Bit No.	Meaning
0...1	Not used
2	Error Queue not empty The bit is set when an entry is made in the error queue. If this bit is enabled by the SRE, each entry of the error queue generates a service request. Thus an error can be recognized and specified in greater detail by polling the error queue. The poll provides an informative error message. This procedure is to be recommended since it considerably reduces the problems involved with remote control.
3	Not used.
4	MAV bit (message available) The bit is set if a message is available in the output buffer which can be read. This bit can be used to enable data to be automatically read from the instrument to the controller.
5	ESB bit Sum bit of the event status register. It is set if one of the bits in the event status register is set and enabled in the event status enable register. Setting of this bit indicates a serious error which can be specified in greater detail by polling the event status register.
6	MSS bit (master status summary bit) The bit is set if the instrument triggers a service request. This is the case if one of the other bits of this registers is set together with its mask bit in the service request enable register SRE.
7	Not used.

IST Flag and Parallel Poll Enable Register (PPE)

As with the SRQ, the IST flag combines the entire status information in a single bit. It can be read by means of a parallel poll (see "Parallel Poll" on page 161) or using the command `*IST`.

The parallel poll enable register (PPE) determines which bits of the STB contribute to the IST flag. The bits of the STB are "ANDed" with the corresponding bits of the PPE, with bit 6 being used as well in contrast to the SRE. The IST flag results from the "ORing" of all results. The PPE can be set using commands *PRE and read using command *PRE?.

6.1.5.4 Event Status Register (ESR) and Event Status Enable Register (ESE)

The ESR is defined in IEEE 488.2. The event status register can be read out using command *ESR on page 166. The ESE is the associated enable part. It can be set using the command *ESE on page 166 and read using the command *ESE?.

Table 6-4: Meaning of the bits used in the event status register

Bit No.	Meaning
0	Operation Complete This bit is set on receipt of the command *OPC exactly when all previous commands have been executed.
1	not used
2	Query Error This bit is set if either the controller wants to read data from the instrument without having sent a query, or if it does not fetch requested data and sends new instructions to the instrument instead. The cause is often a query which is faulty and hence cannot be executed.
3	Device-dependent Error This bit is set if a device dependent error occurs. An error message with a number between 300 and 399 or a positive error number, which denotes the error in greater detail, is entered into the error queue.
4	Execution Error This bit is set if a received command is syntactically correct but cannot be performed for other reasons. An error message with a number between 200 and 300, which denotes the error in greater detail, is entered into the error queue.
5	Command Error This bit is set if a command which is undefined or syntactically incorrect is received. An error message with a number between 100 and 200, which denotes the error in greater detail, is entered into the error queue.
6	User Request This bit is set on pressing the LOCAL key, i.e. when the instrument is switched over to manual operation.
7	Power On (supply voltage on) This bit is set on switching on the instrument.

6.1.5.5 Application of the Status Reporting System

The purpose of the status reporting system is to monitor the status of one or several devices in a measuring system. To do this and react appropriately, the controller must receive and evaluate the information of all devices. The following standard methods are used:

- **Service request** (SRQ) initiated by the instrument
- **Serial poll** of all devices in the bus system, initiated by the controller in order to find out who sent a SRQ and why

- **Parallel poll** of all devices
- Query of a **specific instrument status** by means of commands
- Query of the **error queue**

Service Request

Under certain circumstances, the instrument can send a service request (SRQ) to the controller. Usually this service request initiates an interrupt at the controller, to which the control program can react appropriately. An SRQ is always initiated if one or several of bits 2, 4 or 5 of the status byte are set and enabled in the SRE. Each of these bits combines the information of the error queue or the output buffer. In order to use the possibilities of the service request effectively, all bits should be set to "1" in the enable registers SRE and ESE.

Example:

Use command `*OPC` to generate an SRQ .

`*ESE 1` - set bit 0 of ESE (Operation Complete)

`*SRE 32` - set bit 5 of SRE (ESB).

After its settings have been completed, the instrument generates an SRQ.

The SRQ is the only possibility for the instrument to become active on its own. Each controller program should set the instrument such that a service request is initiated in the case of malfunction. The program should react appropriately to the service request.

Serial Poll

In a serial poll, just as with command `*STB`, the status byte of an instrument is queried. However, the query is realized via interface messages and is thus clearly faster.

The serial poll method is defined in IEEE 488.1 and used to be the only standard possibility for different instruments to poll the status byte. The method also works for instruments which do not adhere to SCPI or IEEE 488.2.

The serial poll is mainly used to obtain a fast overview of the state of several instruments connected to the controller.

Parallel Poll

In a parallel poll, up to eight instruments are simultaneously requested by the controller using a single command to transmit 1 bit of information each on the data lines, i.e., to set the data line allocated to each instrument to a logical "0" or "1".

In addition to the SRE register, which determines the conditions under which an SRQ is generated, there is a Parallel Poll Enable register (PPE) which is ANDed with the STB bit by bit, considering bit 6 as well. This register is ANDed with the STB bit by bit, considering bit 6 as well. The results are ORed, the result is possibly inverted and then sent as a response to the parallel poll of the controller. The result can also be queried without parallel poll using the command `*IST`.

The instrument first has to be set for the parallel poll using the command PPC. This command allocates a data line to the instrument and determines whether the response is to be inverted. The parallel poll itself is executed using PPE.

The parallel poll method is mainly used to find out quickly which one of the instruments connected to the controller has sent a service request. To this effect, SRE and PPE must be set to the same value.

Query of an instrument status

Each part of any status register can be read using queries. There are two types of commands:

- The common commands *ESR?, *IDN?, *IST?, *STB? query the higher-level registers.
- The commands of the STATus system query the SCPI registers (STATus:QUEStionable...)

The returned value is always a decimal number that represents the bit pattern of the queried register. This number is evaluated by the controller program.

Queries are usually used after an SRQ in order to obtain more detailed information on the cause of the SRQ.

Error Queue

Each error state in the instrument leads to an entry in the error queue. The entries of the error queue are detailed plain text error messages that can be looked up in the Error Log or queried via remote control using SYSTem:ERRor[:NEXT]? or SYSTem:ERRor:ALL?. Each call of SYSTem:ERRor[:NEXT]? provides one entry from the error queue. If no error messages are stored there any more, the instrument responds with 0, "No error".

The error queue should be queried after every SRQ in the controller program as the entries describe the cause of an error more precisely than the status registers. Especially in the test phase of a controller program the error queue should be queried regularly since faulty commands from the controller to the instrument are recorded there as well.

6.1.5.6 Reset Values of the Status Reporting System

The following table contains the different commands and events causing the status reporting system to be reset. None of the commands, except *RST and SYSTem:PRESet, influence the functional instrument settings. In particular, DCL does not change the instrument settings.

Table 6-5: Resetting the status reporting system

Event	**Switching on supply voltage** **Power-On-Status-Clear**		**DCL, SDC** **(Device Clear, Selected Device Clear)**	***RST or SYS-Tem:PRE-Set**	**STA-Tus:PRE-Set**	***CLS**
Effect	**0**	**1**				
Clear STB, ESR	-	yes	-	-	-	yes
Clear SRE, ESE	-	yes	-	-	-	-
Clear PPE	-	yes	-	-	-	-
Clear error queue	yes	yes	-	-	-	yes
Clear output buffer	yes	yes	yes	1)	1)	1)
Clear command processing and input buffer	yes	yes	yes	-	-	-
1) The first command in a command line that immediately follows a <PROGRAM MESSAGE TERMINATOR> clears the output buffer.						

6.1.6 General Programming Recommendations

Initial instrument status before changing settings

Manual operation is designed for maximum possible operating convenience. In contrast, the priority of remote control is the "predictability" of the instrument status. Thus, when a command attempts to define incompatible settings, the command is ignored and the instrument status remains unchanged, i.e. other settings are not automatically adapted. Therefore, control programs should always define an initial instrument status (e.g. using the *RST command) and then implement the required settings.

Command sequence

As a general rule, send commands and queries in different program messages. Otherwise, the result of the query may vary depending on which operation is performed first (see also Preventing Overlapping Execution).

Reacting to malfunctions

The service request is the only possibility for the instrument to become active on its own. Each controller program should instruct the instrument to initiate a service request in case of malfunction. The program should react appropriately to the service request.

Error queues

The error queue should be queried after every service request in the controller program as the entries describe the cause of an error more precisely than the status registers. Especially in the test phase of a controller program the error queue should be queried regularly since faulty commands from the controller to the instrument are recorded there as well.

7 Remote Control Commands

In the following, all remote-control commands will be presented in detail with their parameters and the ranges of numerical values.

7.1 Common Commands

Common commands are described in the IEEE 488.2 (IEC 625-2) standard. These commands have the same effect and are employed in the same way on different devices. The headers of these commands consist of "*" followed by three letters. Many common commands are related to the Status Reporting System.

Available common commands:

***CAL?**

Calibration Query

Initiates a calibration of the instrument and subsequently queries the calibration status. Responses > 0 indicate errors.

Usage: Query only

***CLS**

CLear Status

Sets the status byte (STB), the standard event register (ESR) and the EVENt part of the QUEStionable and the OPERation registers to zero. The command does not alter the mask and transition parts of the registers. It clears the output buffer.

Usage: Setting only

***ESE** <Value>

Event Status Enable

Sets the event status enable register to the specified value. The query returns the contents of the event status enable register in decimal form.

Parameters:

<Value> Range: 0 to 255

***ESR?**

Event Status Read

Returns the contents of the event status register in decimal form and subsequently sets the register to zero.

Return values:

<Contents> Range: 0 to 255

Usage: Query only

***IDN?**

IDeNtification: returns the instrument identification.

Return values:

<ID> "Rohde&Schwarz,<device type>,<serial number>,<firmware version>"

Example: `Rohde&Schwarz,ZVA8-4Port,12345,0.10.1.23`

Usage: Query only

***IST?**

Individual STatus query

Returns the contents of the IST flag in decimal form. The IST flag is the status bit which is sent during a parallel poll.

Return values:

<ISTflag> 0 | 1

Usage: Query only

***OPC**

OPeration Complete

Sets bit 0 in the event status register when all preceding commands have been executed. This bit can be used to initiate a service request. The query form writes a "1" into the output buffer as soon as all preceding commands have been executed. This is used for command synchronization.

***OPT?**

OPTion identification query

Queries the options included in the instrument. For a list of all available options and their description refer to the CD-ROM.

Return values:

<Options>	The query returns a list of options. The options are returned at fixed positions in a comma-separated string. A zero is returned for options that are not installed.
Usage:	Query only

***PCB** <Address>

Pass Control Back

Indicates the controller address to which remote control is returned after termination of the triggered action.

Setting parameters:

<Address>	Range:	0 to 30
Usage:	Setting only	

***PRE** <Value>

Parallel poll Register Enable

Sets parallel poll enable register to the indicated value. The query returns the contents of the parallel poll enable register in decimal form.

Parameters:

<Value>	Range:	0 to 255

***PSC** <Action>

Power on Status Clear

Determines whether the contents of the `ENABle` registers are preserved or reset when the instrument is switched on. Thus a service request can be triggered when the instrument is switched on, if the status registers ESE and SRE are suitably configured. The query reads out the contents of the "power-on-status-clear" flag.

Parameters:

<Action>	0 | 1 **0** The contents of the status registers are preserved. **1** Resets the status registers.

***RCL** <Number>

ReCaLl calls up the instrument settings from an intermediate memory identified by the specified number. The instrument settings can be stored to this memory using the command `*SAV` with the associated number.

It also activates the instrument settings which are stored in a file and loaded using the `MMEMory:LOAD <number>, <file_name.extension>` command.

***RST**

ReSeT

Sets the instrument to a defined default status. The default settings are indicated in the description of commands.

The command is equivalent to `SYSTem:PRESet`.

Usage: Setting only

***SAV** <Number>

SAVe stores the current instrument settings under the specified number in an intermediate memory. The settings can be recalled using the command `*RCL` with the associated number.

To transfer the stored instrument settings in a file, use the command `:MMEMory:STORe:STATe`.

***SRE** <Contents>

Service Request Enable

Sets the service request enable register to the indicated value. This command determines under which conditions a service request is triggered.

Parameters:

<Contents>	Contents of the service request enable register in decimal form. Bit 6 (MSS mask bit) is always 0. Range: 0 to 255

***STB?**

STatus Byte query

Reads the contents of the status byte in decimal form.

Usage: Query only

***TRG**

TRiGger

Triggers all actions waiting for a trigger event. In particular, *TRG generates a manual trigger signal (Manual Trigger). This common command complements the commands of the `TRIGger` subsystem.

Usage: Event

***TST?**

self TeST query

Triggers selftests of the instrument and returns an error code in decimal form (see Service Manual supplied with the instrument). "0" indicates no errors occured.

Usage: Query only

***WAI**

WAIt to continue

Prevents servicing of the subsequent commands until all preceding commands have been executed and all signals have settled (see also command synchronization and `*OPC`).

Usage: Event

7.2 R&S Signal Generator Waveform and List Format

The R&S WinIQSIM2 uses a tag-oriented format for externally or internally generated waveforms as well as for data and control lists. The data can be transmitted via the GPIB interface from an external computer to the instrument and vice versa.

A waveform version number is not necessary due to the flexible, tag-based form.

Tags are self-contained information units. Their general format is `{Name: Data}` or `{Name-Length: Data}`. The colon separates the name part and the data part. The colon can be followed by a space for the sake of legibility.

The **Name** identifies the tag. It is always expressed in capital letters.

The **Data** is tag-specific, and in most cases it is in ASCII plain text.

The **Length** specifies the number of bytes in a WAVEFORM tag or DATA LIST tag (from **:** to }, ASCII integer).

Each waveform file must begin with the TYPE tag. The sequence of the remaining tags is arbitrary. For each tag an indication shows whether it must be included in the file concerned (mandatory) or may be included (optional).

Unknown tags are not analyzed by the R&S WinIQSIM2. On the other hand they are left unchanged, are saved without an error message and can be read back again.

R&S AMIQ waveforms can also be loaded on the instrument, where they are converted internally into an R&S WinIQSIM2 waveform.

7.2.1 Tag Description

This section first describe the mandatory TYPE tag which identifies the file and is always located at the start of the file. The rest of the tags used in the R&S WinIQSIM2 are then described in alphabetical order. Most tags are valid for all three file types. If a tag is valid only for a single file type, e.g. only for a waveform, this fact is indicated in the description.

{TYPE: magic, xxxxxxxx}

(mandatory, must be the first tag in the file)

The `TYPE` tag identifies the file as a valid R&S WinIQSIM2 file. It must be present and must be the first in the waveform. If a file of the same name already exists on the target medium, it is overwritten.

Tip: R&S AMIQ waveforms can also be loaded on the instrument, where they are converted internally into an R&S WinIQSIM2 waveform.

Setting parameters:

magic

Designates the file type and has the following values:

SMU-WV
The file contains a valid R&S WinIQSIM2 waveform.

SMU-MWV
The file contains a valid R&S WinIQSIM2 multi segment waveform.

SMU-DL
The file contains a valid R&S WinIQSIM2 data list.

SMU-CL
The file contains a valid R&S WinIQSIM2 control list.

xxxxxxxx

Is an ASCII-coded checksum of the data part of the `WAVEFORM` tag in the file. This value is always 0 for data lists and control lists.
The checksum for waveforms is used for detecting transmission errors. If the `TYPE` tag contains 0 or a non-numeric value for the checksum, it is ignored.
It is calculated in accordance with the algorithm given bellow, where 'start' is a pointer to the first byte after the '#' character in the `WAVEFORM` tag and 'length' is the number of bytes between 'start' and the closing curly bracket (excluding the latter; 'length' must be divisible by 4 without a remainder):

```
UINT32 checksum(void *start, UINT32 length)
{
UINT32 i, result = 0xA50F74FF;
for(i=0; i < length/4; i++)
result = result ^ ((UINT32 *)start)[i];
return(result);
}
```

Example:

```
{TYPE: SMU-WV,106656}
BB:ARB:WAV:TAG? 'TYPE'
```

queries the content of the `TYPE` tag.

```
Response: 'SMU-WV,106656'
```

this is a valid waveform.

{CLOCK: frequency}

(mandatory for waveforms)

The tag specifies the clock frequency at which the waveform has to be output, in Hz (on multi segment waveforms this tag contains the maximal clock of all segments).

A query of `ARB:CLOCk?` after loading the waveform returns the value set using the `CLOCK` tag. This value can subsequently be altered with the aid of the `ARB:CLOCk?` command.

Example: `{CLOCK: 54000000}`
`BB:ARB:WAV:TAG? 'CLOCK'`
queries the content of the `CLOCK` tag.
`Response: 54000000`
the clock frequency is set to 54 MHz.

Usage: Setting only

{COMMENT: string}

The tag contains a plain text ASCII string of arbitrary length. The string is not analyzed in the R&S WinIQSIM2. It is used to describe the file. The string is allowed to contain all printable ASCII characters except the closing curly bracket.

Example: `{COMMENT: File with data for 3GPP enhanced channels}`
`BB:ARB:WAV:TAG? 'COMMENT'`
queries the content of the `COMMENT` tag of the selected waveform file.
`Response: 'File with data for 3GPP enhanced channels'`
the comment on the waveform reads "File with data for 3GPP enhanced channels".

Usage: Setting only

{COPYRIGHT: string}

The tag contains an ASCII string of arbitrary length. The string is not analyzed in the R&S WinIQSIM2. It is used to store copyright information about the file content.

Example: `{COPYRIGHT: Rohde&Schwarz}`
`BB:ARB:WAV:TAG? 'COPYRIGHT'`
queries the content of the `COPYRIGHT` tag of the selected waveform file.
`Response: 'Rohde&Schwarz'`
copyright resides with Rohde&Schwarz.

Usage: Setting only

{DATA BITLENGTH: BitLength}

(mandatory for data lists)

The tag contains the length of the data held in the `DATA LIST` tag in bits in ASCII format.

Example: `{DATA BITLENGTH: 444}`
`BB:DM:DLIS:TAG? 'DATA BITLENGTH'`
queries the content of the `DATA BITLENGTH` tag of the selected data list file.
`Response: '444'`
the data list is 444 bits long.

Usage: Setting only

{DATA LIST-Length: #d0d1...dx...dN-1…}

(mandatory for data lists)

The tag contains the actual bit sequence of the data list in binary format.

Setting parameters:

Length — Defines the number of bytes in the `DATA LIST` tag in ASCII Format (for format see `WAVEFORM` length).

dx — Data bits in binary format (8-bit unsigned characters, MSB first).

Example: `DATA LIST-17: #d0d1...dx...d127}`
16 bytes containing 128 data bits, first bit is the MS bit of the first byte.

Usage: Setting only

{DATE: yyyy-mm-dd;hh:mm:ss}

(optional)

The tag contains the date and time at which the file was created. The year must be expressed as four digits. The instrument does not analyze this tag.

Example: `{DATE: 2009-04-02;14:32:12}`
`BB:ARB:WAV:TAG? 'DATE'`
queries the content of the `DATE` tag of the selected waveform file.
`Response: '2009-04-02;14:32:12'`
the waveform was created on April 2, 2009 at 14 hrs 32 min

Usage: Setting only

{CONTROL LENGTH: ControlLength}

(optional / recommended for marker and control lists)

The tag contains the length of the control list in ASCII format. If this tag is not used, the marker and control list length is determined by the highest position used in one of the `LIST` tags. It is recommended to set this value to be equal to the number of samples in order to keep marker signals and wave data synchronized.

Example:

```
{CONTROL LENGTH: 444}
BB:ARB:WAV:TAG? 'CONTROL LENGTH'
```

queries the length of the control list.

```
Response: '444'
```

Usage: Setting only

{LEVEL OFFS: RMSOffset_dB,PeakOffset_dB}

(recommended for waveforms)

The tag determines the level of the ARB signal in the waveform file. The offset levels define the offset of rms and peak value relative to the 16-bit full scale modulation (-32767 to + 32767) = 0 dB.

Setting parameters:

RMSOffset_dB
Defines the RMS level offset of the signal relative to full scale ARB signal in the `WAVEFORM` tag. The offset is defined in ASCII float format. The value is always positive.
A 3dB value indicates that the RMS level of the signal is 3 dBs below the full scale.
full scale = max. amplitude of vector of I/Q samples = $|S_{IQ}|$max = sqrt(I^2+Q^2)max = 0 dB

PeakOffset_dB
Defines the peak level offset of the signal relative to full scale for the ARB signal in the `WAVEFORM` tag. The offset is defined in ASCII float format.
The value usually equals 0 dB as in the majority of cases the I/Q samples (signed 16-bit integer values) are modulated to full scale:
Full scale = 0 dB = max. amplitude of vector of I/Q samples = $|S_{IQ}|$max = sqrt(I^2+Q^2)max = (2^{15})-1 = 32767.
A positive `PeakOffset_dB` value indicates that a headroom to full scale is provided when generating the waveform. A negative PeakOffset_dB value indicates that overrange is likely for some samples, i.e. clipping might occur.
The crest factor can be calculated from the two values as follows:

```
Crest Factor = |PeakOffset_dB - RMSOffset_dB|
```

Example:

```
{LEVEL OFFS: 3.45,2}
BB:ARB:WAV:TAG? 'LEVEL OFFS'
```

queries the content of the 'LEVEL OFFS' tag of the selected waveform file.

```
Response: '3.45,2'
```

the level of the waveform is below full scale, clipping will not occur.

Usage: Setting only

{SAMPLES: Samples}

(recommended for waveforms)

The tag contains the number of I/Q samples in the waveform in ASCII format.

On multi segment waveforms, this tag contains the total I/Q samples of all segments.

Example:

```
{SAMPLES: 4333}
BB:ARB:WAV:TAG? 'SAMPLES'
```

queries the content of the `SAMPLES` tag of the selected waveform file.

```
Response: '4333'
```

the waveform contains 4333 I/Q samples.

Usage: Setting only

{[TRACE] LIST [#]: Pos0:Val0;Pos1:Val1;...Posx:Valx;...PosN-1:ValN-1}

(mandatory for control lists / optional for waveforms)

The tag contains the data for the marker and control signals in the control list or the marker signals of ARB waveforms. To select which of these signals is defined, `[TRACE]` and the associated number are used. For ARB waveforms it is only meaningful to define marker signals (on ARB multi segment waveforms these tags will be ignored).

Setting parameters:

[TRACE] Specifies the name of the marker or control signal.
You may choose from the following names:
`MARKER; BURST; LEVATT; CW MODE; HOP, MAP`

[#] Specifies the number in the case of control signals and marker signals with the same name. There is a choice between 4 markers and 3 LEVATT signals. Lines LEVATT 1 and 2 are needed for internal purposes and should not be used.

Pos Specifies in ASCII format the number of the position in the sequence, with effect from which the binary state of the marker or of the control signal changes from 0 to 1 or from 1 to 0.

Val Specifies the binary state of the marker or of the control signal {0; 1} from Pos_x to Pos_{x+1} exclusive in ASCII format.

Example:

```
{MARKER LIST 1: 0:0;10:1;20:0;30:1}
BB:DM:CLIS:TAG? 'MARKER LIST 1'
```

queries the content of the `MARKER LIST 1` tag of the selected control list file.

```
Response: '0:0;10:1;20:0;30:1'
```

the marker setting for samples 0 to 9 = 0 (low), for 10 to 19 = 1 (high) and for 20 to 29 = 0. From sample 30 onward the marker setting = 1.

Example:

```
{LEVATT LIST 1: 0:0;10:1;20:0;30:1}
BB:DM:CLIS:TAG? 'LEVATT LIST 1'
```

queries the content of the `LEVATT LIST 1` tag of the selected control list file.

```
Response: '0:0;10:1;20:0;30:1'
```

level attenuation applies to data values 10 to 19 (high) and from data value 30 onward.

Usage: Setting only

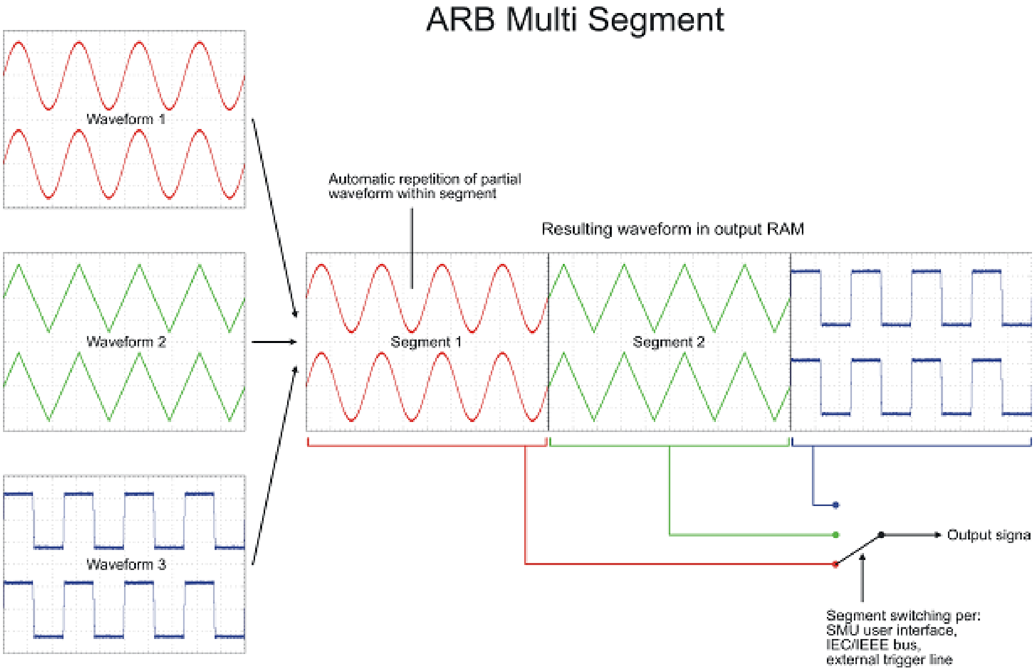
{WAVEFORM-Length: #I0Q0I1Q1...IxQx...IN-1QN-1…}

(mandatory for waveforms)

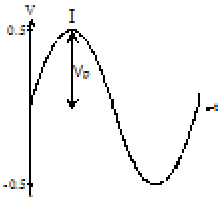
The tag contains the actual waveform data or multi segment waveform data (I/Q stream).

The `TYPE` tag decides, wheather the waveform file contains a normal waveform or a multi segment waveform.

The following figure shows the prinziple of multi segment work mode.



At 50 Ohm between the inner and outer conductors of the I and Q outputs, an amplitude Vp of +/-0.5 Volt arises on the I/Q output connectors of the R&S WinIQSIM2. The illustration also applies to the Q output.



Setting parameters:

Length — Specifies the number of bytes in a `WAVEFORM` tag and is calculated as follow:
`Length` = `Number of I/Q pairs * 4` (i.e. 2 bytes per I and 2 bytes per Q value) + `1 byte` (i.e. the length of the `#`).

I Q

IxQx… represents binary data (16-bit signed integer in 2's complement notation) containing the I and Q component alternately and starting with the I component. Each component consists of two bytes, least significant byte (LSB) first.
The values of the two bytes in an I component and a Q component are in the range 0x0 to 0xFFFF (-32767 to +32767). This value is transferred to the D/A converter.
This tag is also used to store multi segment waveforms. The I/Q streams of the individual waveforms are directly concatenated to one collectively waveform I/Q stream.
The number of segments and the start offset and length of the individual segments inside the total waveform I/Q stream is determined by the additional tags MWV_SEGMENT_COUNT, MWV_SEGMENT_START and MWV_SEGMENT_LENGTH.
Further MWV_SEGMENT_... tags are also available, i.e. for level and clock information etc.

Example:

{WAVEFORM-401:#I_0,Q_0,I_1,Q_1,I_2,Q_2,...I_{99},Q_{99}}
100 I/Q pairs with 4 bytes each are transmitted - none multi segment

Example:

Multi Segment
{WAVEFORM-1201:
#$I_{0,Seg0}$,$Q_{0,Seg0}$,$I_{1,Seg0}$,$Q_{1,Seg0}$,...$I_{99,Seg0}$,$Q_{99,Seg0}$,$I_{0,Seg1}$,$Q_{0,Seg1}$,$I_{1,Seg1}$,
$Q_{1,Seg1}$,... $I_{199,Seg1}$,$Q_{199,Seg1}$}
(2 segments: segment 0 with 100 I/Q pairs, segment 1 with 200 I/Q pairs. Each I/Q pair consist of 2*16 bit = 4 bytes)
The data is transmitted using SCPI command
BB:ARB:WAVE:DATA.

Usage: Setting only

{MWV_SEGMENT_COUNT: NumOfSeg}

(mandatory for multi segment waveforms)

The tag contains the number of segments in the multi segment waveform in ASCII integer format.

Example:

```
{MWV_SEGMENT_COUNT: 2}
```

Multi segment waveform with 2 segments

Usage: Setting only

{MWV_SEGMENT_LENGTH: SamplesSeg0, SamplesSeg1, …, SamplesSegN-1}

(mandatory for multi segment waveforms)

The tag contains a list of I/Q sample lengths for every segment in the multi segment waveform in ASCII integer format.

Example: `{MWV_SEGMENT_LENGTH: 100,200}`
2 segments: 100 samples in segment 0 and 200 samples in segment 1.

Usage: Setting only

**{MWV_SEGMENT_START:
SampleStartOffsetSeg0, SampleStartOffsetSeg1, …, SampleStartOffsetSegN-1}**

(mandatory for multi segment waveforms)

The tag contains a list of I/Q sample start offsets for every segment in the multi segment waveform in ASCII integer format.

Example: `{MWV_SEGMENT_START: 0,100}`
2 segments with 100 samples in segment 0 and 200 samples in segment 1.
Sample start offset of first segment 0 is 0 samples, start offset of next segment 1 is the sample length of segment 0 = 100 samples.

Usage: Setting only

{MWV_SEGMENT_CLOCK_MODE: Mode}

(mandatory for multi segment waveforms)

The tag contains a string in ASCII format which supplies the clock rate mode, that was used for calculation of the multi segment output waveform.

The tag `CLOCK` contains always the highest clock rate of all segments.

The tag `MWV_SEGMENT_CLOCK` contains the clock rates of the individual segments.

Setting parameters:

Mode

UNCHANGED
The segments may have different clock rates.
During the multi segment calculation, the clock rates of all individual original waveforms were taken over unchanged in the segments of the multi segment output waveform (no software resampling is done).
If the segments have different clock rates, there are some restrictions on signal output, i.e. switching per external trigger and seamless switching between segments is not possible.
If all segments have already an identical clock rate, this mode is usually set to `HIGHEST`.

HIGHEST
All segments have an indentical clock rate, which is the highest clock rate of all original waveforms.
If an individual original waveform has a lower clock rate, an upsampling to the highest clock rate will performed on calculation of the multi segment output waveform.

USER
All segments have an indentical clock rate, which is given by the user.
Note: Only upsampling is allowed, no downsampling!
If an individual original waveform has a lower clock rate, an upsampling to the user clock rate will performed on calculation of the multi segment output waveform.

Example:

```
{MWV_SEGMENT_CLOCK_MODE: UNCHANGED}
{MWV_SEGMENT_CLOCK_MODE: HIGHEST}
{MWV_SEGMENT_CLOCK_MODE: USER}
```

Usage: Setting only

{MWV_SEGMENT_CLOCK: ClockSeg0, ClockSeg1, …, ClockSegN-1}

(mandatory for multi segment waveforms)

The tag contains a list of clock frequencies for every segment in the multi segment waveform in ASCII floating point format.

Example:

```
{MWV_SEGMENT_CLOCK: 100e6,80e6}
```

2 segments: clock of segment 0 is 100 MHz, clock of segment 1 is 80 MHz.
Note: If the segments have different clock frequencies, there are some restrictions on signal output, i.e. seamless switching between segments is only possible, if all segments have the same clock frequency. Software resampling (upsampling) can be used to bring all segments to the same clock.

Usage: Setting only

{MWV_SEGMENT_LEVEL_OFFS:
RMSOffset_dBSeg0, PeakOffset_dBSeg0, …,RMSOffset_dBSegN-1, PeakOffset_dBSegN-

(mandatory for multi segment waveforms)

The tag contains a list of level pairs in ASCII floating point format, one pair for every segment in the multi segment waveform. The first value of a level pair defines the rms offset and the second value the peak offset relative to the 16-bit full scale modulation (-32767; + 32767) = 0 dB. The meaning of one level value pair is the same as in the LEVEL OFFS tag for normal waveforms.

Example: `{MWV_SEGMENT_LEVEL_OFFS: 3.0,0.0,6.0,0.0}`
2 segments: RMS level of segment 0 is 3dB below full scale; RMS level of segment 1 ist 6dB below full scale. Peak level of both segments is 0 dB full scale.

Usage: Setting only

{MWV_SEGMENT_FILES:
“FileNameSeg0.wv”, “FileNameSeg1.wv”, …, “FileNameSegN-1.wv”}

(optional for multi segment waveforms)

The tag contains a list of file names for every segment in the multi segment waveform in ASCII format.

Example: `{MWV_SEGMENT_FILES: “d:\waveforms\sine.wv”, “d:\waveforms\rect.wv”}`

Usage: Setting only

{MWV_SEGMENTx_COMMENT: text}

(optional for multi segment waveforms)

The tag contains a user comment for a specific segment x = [0…NumOfSeg-1] in the multi segment waveform in ASCII format.

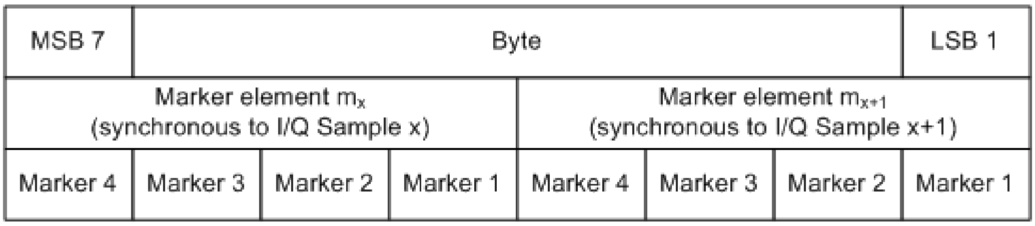
Example: `{MWV_SEGMENT1_FILES: segment 1 contains a QPSK signal.}`

Usage: Setting only

{CONTROL LIST WIDTH4–Length: #m0m1…mx…mM-1}

(optional for waveforms and multi segment waveforms)

The tag contains a binary marker element stream, which will be given out synchronously to the I/Q sample sequence. One marker element mx consists of 4 bit, which are assigned to the 4 possible marker lines of the instrument (one bit per marker line). One 4 bit marker elements is needed for every I/Q sample in the `WAVEFORM` tag - so the number of marker elements M should be equal to the number of I/Q samples. The `CONTROL LENGTH` tag has to contain the number of all marker elements M.



MSB 7	Byte						LSB 1
Marker element m_x (synchronous to I/Q Sample x)				Marker element m_{x+1} (synchronous to I/Q Sample x+1)			
Marker 4	Marker 3	Marker 2	Marker 1	Marker 4	Marker 3	Marker 2	Marker 1

Fig. 7-1: Marker element in 4-bit binary format bit order

For standard waveforms the `MARKER LIST x` tags are a more compact way to define markers, but in prinziple this `CONTROL LIST WIDTH4` format can also be used instead of the `MARKER LIST x` tags.

For multi segment waveforms this `CONTROL LIST WIDTH4` format is required for marker definition. The binary marker streams of the individual segments are directly concatenated (without any gap) to one collectively marker stream.

Setting parameters:

Length
Defines the number of bytes in the `CONTROL LIST WIDTH4` tag in ASCII Format and is calculated as follow:
Length = Size of "#" (1 byte) + Number of marker elements m_x * (4 bit) / (8 bits/byte)
The value is rounded up for byte alignment.

mx
Marker element in 4-bit binary format.

Example: {CONTROL LIST WIDTH4-51: #$m_0m_1...m_x...m_{99}$}
100 marker elements, each marker element with 4 bits

Usage: Setting only

7.2.2 Creating a Waveform Manually

We will use the example of a sine function in the I channel and a cosine function in the Q channel, each with 20 points, to explain how a waveform file `SICO.WV` is generated.

The sine and cosine values are calculated by a short program written in the programming language C (see the following example for creating a C-program). They are stored in the file `SICO.TXT` as follows:

```
Contents of SICO.TXT:
Sine (I)  Cosine (Q)

0.000000 1.000000
0.309017 0.951057
0.587785 0.809017
0.809017 0.587785
0.951057 0.309017
1.000000 -0.000000
0.951056 -0.309017
0.809017 -0.587785
0.587785 -0.809017
```

```
0.309017 -0.951056
-0.000000 -1.000000
-0.309017 -0.951057
-0.587785 -0.809017
-0.809017 -0.587785
-0.951056 -0.309017
-1.000000 0.000000
-0.951056 0.309017
-0.809017 0.587785
-0.587785 0.809017
-0.309017 0.951057
```

The decimal values in SICO.TXT should be normalized such that they are between –1.0 and +1.0. The waveform file SICO.WV will be based on the contents of this file.

To be read by the R&S WinIQSIM2 these waveform data must be coded binary and packed into an appropriate WAVEFORM tag. A tag consists of a name and a data set and is enclosed in curved brackets. The tag is a kind of label carrying information about what the R&S WinIQSIM2 should do with the data set (see above and step 3 of the following instructions).

The following steps outline how to create the waveform file SICO.WV:

1. The values from the file SICO.TXT must be converted into binary format consisting of integer numbers without a sign and with 16-bit width. The numeric range between –1.0 and +1.0 corresponds to the modulation range of the waveform D/A converter of 65535 (2^{16}-1).

+1.0 ->	32767	= 0x7FFF
0.0 ->	0	= 0x0000
-1.0 ->	-32767	= 0x8001

   A further C-program is suitable for creating the binary data set from the ASCII values stored in SICO.TXT file (see following example for creating a C-program). This program stores the binary data set to a file called SICO.WV.

   The contents of the file SICO.WV reads as follows:

   ```
   IQIQIQIQIQIQIQI ... IQ
   ```

   Note: There is no readable representation for binary values in this document. This is why we use the sequence IQIQIQ to characterize the binary code in the present example.

2. The file SICO.WV now contains the binary data set corresponding to the 20 I/Q pairs. Before this binary data set can be further processed in step 3, the TYPE and the CLOCK tag must be placed in front.
   To enter the TYPE tag in the SICO.WV file an ASCII editor which is able to handle binary data as well, e.g. the Microsoft Windows editor NOTEPAD.EXE, must be used.

   Now the contents of the SICO.WV file read:
   ```
   {TYPE: SMU-WV, 0}{CLOCK: 10e6} IQIQIQIQIQIQIQIQIQI ... IQ
   ```

To simplify our example the checksume is set to 0, i.e. the instrument does not evaluate a checksum.

3. The binary data must now be packed into a WAVEFORM tag with the following structure:
   `{WAVEFORM-Length: #IQIQIQIQIQIQIQIQIQI ... IQ}`
   The editor mentioned above which can handle binary data is now used to place the string `{WAVEFORM-81: #` in front and the symbol `}` at the end of the data set.
   The `Length` of the data set is the number of bytes of the data set and is calculated as the sum of the number of I/Q pairs * 4 (2 bytes per I and per Q value) and one 1 byte for the symbol `#`.
   In our example containing a sine and a cosine with 20 pairs for each wave, the resulting length is 81.

   Note: There is no readable representation for binary values in this document. This is why we use the sequence `IQIQIQ` to characterize the binary code in the present example!

   The contents of the waveform file `SICO.WV` for 20 I/Q pairs is now ready for operation and reads.
   ```
   {TYPE: SMU-WV,0}
   {CLOCK: 10e6}
   {WAVEFORM-81: #IQIQIQIQIQIQ ... IQ}
   ```
   The tags `TYPE`, `CLOCK` and `WAVEFORM` are mandatory for each waveform. All other tags are optional and can be inserted after the `TYPE` tag in arbitrary order, e.g.
   ```
   {TYPE: SMU-WV,0}
   {COMMENT: I/Q=sine/cosine, 20 points, clock 10 MHz}
   {CLOCK: 10e6}
   {Samples:20}
   {WAVEFORM-81:#IQIQIQIQIQIQ ... IQ}
   ```

C-programs for creating the SICO.TXT file and for generating the waveform file

C-program for creating the file `SICO.TXT` containing 20 sine and cosine pairs:

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
void main(void)
{
    const unsigned int samples = 20;
    const float pi = 3.141592654;
    float grad, rad;
    FILE *fp;
    int i;
    fp = fopen("SICO.TXT", "w");
    if (fp == 0)
        return;
    for (i=0; i<samples; i++)
    {
        grad = (360.0 / (float)samples) * (float)i;
```

```
        rad = grad * (pi / 180.0);
        fprintf(fp, "%f %f\n", sin(rad), cos(rad));
    }
    fclose(fp);
}
```

Extract from a C-program generating a binary data set from the I/Q pairs in the file `SICO.TXT` and storing the result to file `SICO.WV`:

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
void main(void)
{
    const unsigned int samples = 20;  // Number of samples to convert
    FILE *fp_sour, *fp_dest;
    float i_float, q_float;
    unsigned short i_usint, q_usint;
    int i;
    fp_sour = fopen("SICO.TXT", "rt");
    if (fp_sour == 0)
        return;
    fp_dest = fopen("SICO.WV", "wb");
    if (fp_dest == 0)
    {
        fclose(fp_sour);
        return;
    }
    // Write required tags to waveform file
    fprintf(fp_dest, "{TYPE: SMU-WV,0}");
    fprintf(fp_dest, "{CLOCK: 10e6}");
    fprintf(fp_dest, "{SAMPLES: %d}", samples);
    fprintf(fp_dest, "{WAVEFORM-%d:#", (samples * 4) + 1);
    for (i=0; i<samples; i++)
    {
        // Read I/Q pair from ASCII file
        if (fscanf(fp_sour, "%f %f", &i_float, &q_float) == EOF)
            break;
        // Convert I/Q pair to unsigned short
        i_usint = (unsigned short)floor((i_float * 32767.0) + 0.5);
        q_usint = (unsigned short)floor((q_float * 32767.0) + 0.5);
        // Write converted I/Q pair to waveform file
        fwrite(&i_usint, 2, 1, fp_dest);
        fwrite(&q_usint, 2, 1, fp_dest);
    }
    fprintf(fp_dest, "}");
    fclose(fp_dest);
    fclose(fp_sour);
}
```

7.2.3 Creating an ASCII C-List Manually

When you create the C-list in the editor, as shown in the figure, you get an ASCII file with the extension `*.dm_iqc` (see the example C-List ASCII File).

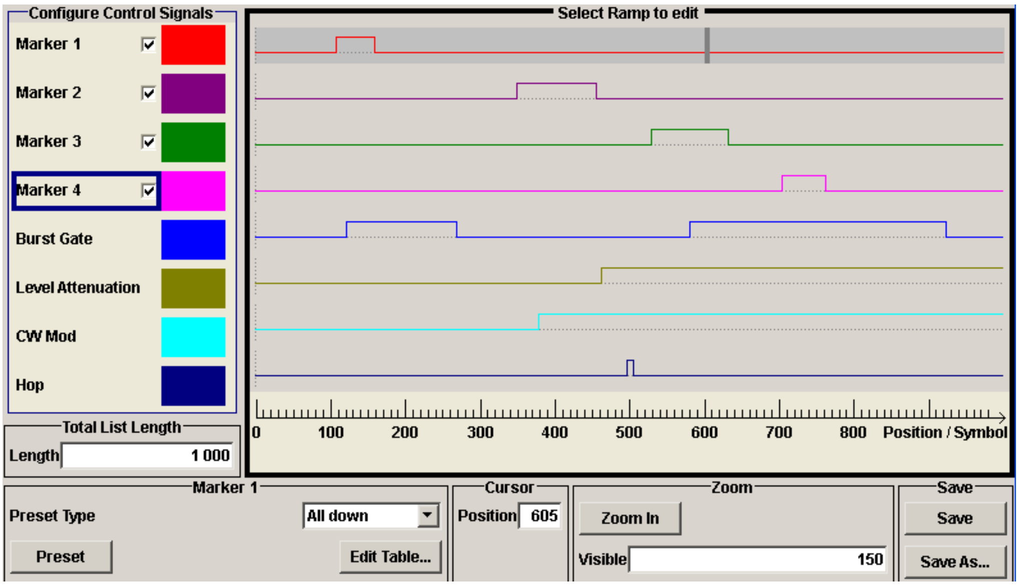


Fig. 7-2: Representation in the C-List Editor

Example: C-List ASCII File

C-List in ASCII format, corresponding to C-List editor figure 7-2:

```
{TYPE:SMU-CL}{COPYRIGHT:2003 Rohde&Schwarz SMU}{DATE:2011-07-19;14:58:44}
{HOP LIST:0:0;498:1;506:0}{CW MODE LIST:0:0;380:1}{LEVATT LIST 1:0:0;464:1}
{BURST LIST:0:0;122:1;270:0;582:1;924:0}
{MARKER LIST 4:0:0;706:1;764:0}{MARKER LIST 3:0:0;530:1;633:0}
{MARKER LIST 2:0:0;350:1;457:0}{MARKER LIST 1:0:0;108:1;160:0}
{CONTROL LENGTH:1000}
```

The tags to the traces in the C-List editor are:

- HOP LIST
- CW MODE LIST
- LEVATT LIST 1
- BURST LIST
- MARKER LIST 4
- MARKER LIST 3
- MARKER LIST 2
- MARKER LIST 1
- CONTROL LENGTH

CONTROL LENGTH defines the periodicity of the total C-List. The remaining tags define the individual markers or control traces in a combined state/starting time way within the C-List period.

If you want to write an ASCII C-List directly without using the C-List editor, a special syntax has to be taken into account:

- The tag and the following [state: start time]-pairs are separated by a colon ":".
- The [state: start time] pairs in turn are separated by semicolons ";".
- The list state can be either 0 or 1.
- The time unit is the simple period.

Example:

"CW MODE LIST 0:0;380:1" for a CW trace:

The list status starts at time 0 with 0, and at time point 380 symbol periods after start, it changes to 1.

With the commands `BB:DM:CLISt:...` you can also generate control lists in binary fomat, if required. See chapter 7.8.2, "DM Lists", on page 244.

7.3 MMEMory Subsystem

The MMEMory subsystem (**M**ass **Mem**ory) contains the commands for managing files and directories as well as for loading and storing complete instrument settings in files.

The various drives can be selected using the "mass storage unit specifier " <msus>. The internal hard disk is selected with `D:\`, and a memory stick which is inserted at the USB interface is selected with `E:\`. The resources of a network can also be selected with <msus> in the syntax of the respective network, e.g. using the UNC format (Universal Naming Convention): `\\server\share`.

The default drive is determined using the command `MMEMory:MSIS <msus>`.

The `C:` drive is a protected system drive. This drive should not be accessed. Reconstruction of the system partition will not be possible without loss of data.

7.3.1 File Naming Conventions

To enable files in different file systems to be used, the following file naming conventions should be observed.

The file name can be of any length and no distinction is made between uppercase and lowercase letters. The file and the optional file extension are separated by a dot. All letters and numbers are permitted (numbers are, however, not permitted at the beginning of the file name). Where possible, special characters should not be used. Use of the slashes "\" and "/" should be avoided since they are used in file paths. A number of names are reserved by the operating system, e.g. `CLOCK$`, `CON`, `AUX`, `COM1...COM4`, `LPT1...LPT3`, `NUL` and `PRN`.

In the R&S Signal Generator all files in which lists and settings are stored are given a characteristic extension. The extension is separated from the actual file name by a dot (see chapter 7.3.2, "Extensions for User Files", on page 187 for an overview the file types).

The two characters "*" and "?" function as "wildcards", i.e. they are used for selecting several files. The "?" character represents exactly one character, while the "*" character represents all characters up to the end of the file name. "*.*" therefore stands for all the files in a directory.

When used in conjunction with the commands, the parameter `<file_name>` is specified as a string parameter with quotation marks. It can contain either the complete path including the drive, only the path and file name, or only the file name. The file name must include the file extension. The same applies for the parameters `<directory_name>` and `<path>`.

Depending on how much information is provided, either the values specified in the parameter or the values specified with the commands `MMEM:MSIS` (default drive) and `MMEM:CDIR` (default directory) are used for the path and drive setting in the commands.

Before the instrument settings can be stored in a file, they have to be stored in an intermediate memory using common command `*SAV <number>`. The specified number is subsequently used in the `:MMEMory:STORe:STATe` command. Also, subsequently to loading a file with instrument settings with command `:MMEMory:LOAD:STATe`, these settings have to be activated with the common command `*RCL <number>`.

7.3.2 Extensions for User Files

The following table list all available file extensions for user files. The currently available files on the instrument depends on the installed options.

7.3.3 Examples

In these examples, the current instrument setting is stored in the file `test1.savrcl` in the directory `user`.

Storing and Loading Current Settings

1. Store the current setting in an intermediate memory with number 4. This setting can be called using command `*RCL` and the associated number of the memory, e.g. `*RCL 4`.
   ```
   *SAV 4
   ```
2. To store the settings in a file in a specific directory, specify the complete path.
   ```
   MMEM:STOR:STAT 4,"d:\user\test.savrcl"
   ```
3. To store the settings in a file in the default drive, set the default drive and specify only the file name.
   ```
   MMEM:MSIS 'D:\user'*SAV 4
   MMEM:STOR:STAT 4,"test.savrcl"
   ```

4. Load the file `test.savrcl` in the `user` directory.
   ```
   MMEM:LOAD:STAT 4,'D:/user/test.savrcl'
   ```

5. Activate the instrument setting of the file `test.savrcl`.
   ```
   *RCL 4
   ```

Working with Files and Directories

1. Read out all files in the specified directory.
   ```
   MMEM:CAT? 'E:/user'
   ```
   Response: `127145265,175325184,"test,DIR,0","temp,DIR,0","readme.txt,ASC,1324","state.savrcl,STAT,5327","waveform.wv,BIN,2342"`

   the directory `E:/user` contains the subdirectories `test` and `temp` as well as the files `readme.txt`, `state.savrcl` and `waveform.wv` which have different file types.

   Tip: To query only the subdirectories of the current or specified directory, perform:
   ```
   MMEM:DCAT? 'E:/user'
   ```
   Response: `'test', 'temp'`

   To query only the number of subdirectories in the current or specified directory, perform:
   ```
   MMEM:DCAT:LENG? 'E:/user'
   ```
   Response: `2`

2. To query the number of files in the current or specified directory, perform:
   ```
   MMEM:CAT:LENG? 'E:/user'
   ```
   Response: `3`

3. Create a new subdirectory for mass memory storage in the specified directory.
   ```
   MMEM:MDIR 'E:/user/new'
   ```

4. Copy the file `state` to a new file.
   ```
   MMEM:COPY 'D:/user/state.savrcl','E:/user/new'
   ```

5. Rename the file `state`.
   ```
   MMEM:MOVE 'state.savrcl','state_new.savrcl'
   ```

6. Remove the `test` directory.
   ```
   MMEM:RDIR 'E:/user/test'
   ```

7.3.4 Remote Control Commands

:MMEMory:CATalog? <path>

Returns the content of the current or a specified directory.

Query parameters:

<path> string

String parameter to specify the directory. If the direcory is omitted, the command queries the content of the current directory, queried with `MMEM:CDIR` command.

Return values:

<path> <used_memory>,<free_memory>,<file_name>,<file_entry>,...

<used_memory>
Total amount of storage currently used in the directory, in bytes.

<free_memory>
Total amount of storage available in the directory, in bytes.

<file_entry>
All files of the directory are listed with their file name, format and size in bytes.

Example: See "Working with Files and Directories" on page 188.

Usage: Query only

:MMEMory:CATalog:LENGth?

Returns the number of files in the current or in the specified directory.

Return values:

<Length> <path>

String parameter to specify the directory. If the direcory is omitted, the command queries the content of the current directory, queried with `MMEM:CDIR` command.

<count> Number of files

Example: See "Working with Files and Directories" on page 188.

Usage: Query only

:MMEMory:CDIRectory <Cdirectory>

Changes the default directory for mass memory storage. The directory is used for all subsequent `MMEM` commands if no path is specified with them. It is also possible to change to a higher directory using two dots '..' .

Setting parameters:

<Cdirectory> <directory_name>

Example: See "Working with Files and Directories" on page 188.

Usage: Setting only
SCPI confirmed

:MMEMory:COPY <file_source>[,<file_destination>]

Copies an existing to a new file. Instead of just a file, this command can also be used to copy a complete directory together with all its files.

It is also possible to specify the path using another parameter. The command is:

```
MMEMory:COPY
<file_source><msus_source>[,<file_destination>,
<msus_destination>]
```

Setting parameters:

<file_source> string

String parameter to specify the name of the file to be copied.
If `<destination>` is not specified, `<source>` is copied to the `MMEM:MSIS` drive and the `MMEM:CDIR` directory. Files which already exist with the same name in the destination directory are overwritten without an error message.
It is also possible to specify the path using another parameter. The command is: `MMEMory:COPY <file_source><msus_source>[,<file_destination>, <msus_destination>]`

file_destination string

String parameter to specify the name of the new file. If no file destination is specified, the source file is copied to the current direcotry, queried with the `MMEM:CDIR` command. Files which already exist with the same name in the destination directory are overwritten without an error message.

Example: See "Working with Files and Directories" on page 188.

Usage: Event
SCPI confirmed

:MMEMory:DATA <file_name>[,<binary block data>]
:MMEMory:DATA? <file_name>

Writes the block data <binary block data> to the file identified by <file_name>. The IEC/IEEE-bus terminator should be set to EOI in order to ensure correct data transfer.

The associated query command transfers the specified file from the instrument to the IEC/IEEE bus and then on to the control computer. It is important to ensure that the intermediate memory on the control computer is large enough to take the file. In this case, the setting for the IEC/IEEE-bus terminator is irrelevant. This command can be used to read/transfer stored instrument settings or waveforms directly from/to the instrument.

Parameters:

<file_name> String parameter to specify the name of the file.

Setting parameters:

<binary block data> #<number><length entry><block data>

# always comes first in the binary block
<number> the first digit indicates how many digits the subsequent length entry has
<length entry> indicates the number of subsequent bytes
<binary block data> binary block data for the specified length.

Example:

`MMEM:DATA 'TEST1.WV',#3767<binary data>`
writes the block data to the file `test1.wv`.
The digit "3" indicates a length entry of 3 digits; the digits "767" indicate a length of the binary data in bytes.
`MMEM:DATA? 'TEST1.WV'`
sends the data of the file `Test1.wv` from the instrumenet to the controller in the form of a binary block.

:MMEMory:DCATalog?

Returns the subdirectories of the current or specified directory.

Query parameters:

<path_name> String parameter to specify the directory. If the direcory is omitted, the command queries the content of the current directory, queried with `MMEM:CDIR` command.

Return values:

<Dcatalog> <file_entry>

Names of the subdirectories separated by colons. The first two strings are related to the parent directory.

Example: See "Working with Files and Directories" on page 188.

Usage: Query only

:MMEMory:DCATalog:LENGth? [<path_name>]

Returns the number of subdirectories in the current or specified directory.

Query parameters:
<path_name> String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory, to be queried with `MMEM:CDIR` command.

Return values:
<file_entry_count> Number of parent and subdirectories.

Example: See "Working with Files and Directories" on page 188.

:MMEMory:DELete <Delete>

Removes a file from the specified directory.

Setting parameters:
<Delete> <file_name>

String parameter to specify the name and directry of the file to be removed.

Example: See "Working with Files and Directories" on page 188.

Usage: Event
SCPI confirmed

:MMEMory:LOAD:STATe <sav_rcl_state_number>, <file_name>

This command loads the specified file stored under the specified name in an internal memory.

After the file has been loaded, the instrument setting must be activated using an `*RCL` command.

Setting parameters:
<sav_rcl_state_number> Determines to the specific <number> to be used with the `*RCL` command, e.g. `*RCL 4`.

<file_name> String parameter to specify the file name with extension `*.savrcl`.

Example: See "Storing and Loading Current Settings" on page 187.

Usage: Setting only

:MMEMory:MDIRectory <directory_name>

Creates a new subdirectory for mass memory storage in the specified directory. If no directory is specified, a subdirectory is created in the default directory. This command can also be used to create a directory tree.

Setting parameters:
<directory_name> string

String parameter to specify the new directory.

Example: See "Working with Files and Directories" on page 188.

Usage: Setting only

:MMEMory:MOVE <file_source>, <file_destination>

Moves an existing file to a new location or, if no path is specified, renames an existing file.

It is also possible to specify the path using another parameter. The command is:

```
MMEMory:MOVE

file_source,msus_source[, file_destination, msus_destination]
```

Setting parameters:

<file_source> String parameter to specify the name of the file to be moved.

<file_destination> String parameters to specify the name of the new file.

Example: See "Working with Files and Directories" on page 188.

Usage: Setting only
SCPI confirmed

:MMEMory:MSIS <Msis>

The command sets the drive (or network resource in the case of networks) using `msus` (MSIS = Mass Storage Identification String). This setting is effective for all `MMEMory` commands where the drive is not explicitly specified in the parameter.

Parameters:

<Msis> string

*RST: D:\

Example: See "Storing and Loading Current Settings" on page 187.

Usage: SCPI confirmed

:MMEMory:RDIRectory <Rdirectory>

Removes an existing directory from the mass memory storage system. If no directory is specified, the subdirectory with the specified name is deleted in the default directory.

Setting parameters:

<Rdirectory> string

String parameter to specify the directory to be deleted.

Example: See "Working with Files and Directories" on page 188.

Usage: Setting only

:MMEMory:STORe:STATe <sav_rcl_state_number>, <file_name>

Stores the current instrument setting in the specified file.

The instrument setting must first be stored in an internal memory with the same number using the common command `*SAV`.

Setting parameters:

<sav_rcl_state_number> Corresponds to the specific <number> defined with the `*SAV` command, e.g. `*SAV 4`.

<file_name> String parameter to specify the file name with extension `*.savrcl`.

Example: See "Storing and Loading Current Settings" on page 187.

Usage: Event

7.4 Transmission Commands

This chapter contains the remote control commands to set waveform generation, instrument configuration and transmission of the file containing the waveform.

GENerate:WAVeform:DURation?

Queries the signal duration in seconds of the available samples within the generated waveform file. The value of this parameter is evaluated by parameter `GENerate:WAVeform:SAMPles`.

Return values:

<Duration> float

Example: `GEN:WAV:DUR?`
queries the signal duration in seconds.

Usage: Query only

GENerate:WAVeform:OSAMpling?

Queries the baseband oversampling factor. The value is determined by the content of the generated waveform file. To adjust the oversampling rate, the corresponding parameters defined within the standards must be used, e.g. `SOUR:BB:WLAN:FILT:OSAM`.

Return values:
<Osampling> float

Example: `GEN:WAV:OSAM?`
queries the oversampling factor.

Usage: Query only

GENerate:WAVeform:SAMPles?

Queries the number of available samples within the generated waveform file. It equals to the I and respectively Q sample sequence length and is calculated as follow:

Number of Samples = Number of symbols * Oversampling factor

Return values:
<Samples> float

Example: `GEN:WAV:SAMP?`
queries the number of available samples.

Usage: Query only

GENerate:WAVeform:SRATe?

Queries the sample rate in [Hz], where

Sampling Rate = Number of symbols * Oversampling factor.

Return values:
<Srate> float

Example: `GEN:WAV:SRAT?`
queries the sample rate.

Usage: Query only

INSTruments:GPIB:ADDRess <GPIB_Address#1>,<GPIB_Address#2>,...

Defines the GPIB addresses of the instruments in the device list.

Parameters:
<GPIB_Address> Range: 1 to 30

Example: `INST:GPIB:ADDR 8,9,10`
sets the GPIB addresses of the first three devices in the list.

INSTruments:NAME <SymbName#1>,<SymbName#2>, ...

Defines the symbolic name of the instruments in the device list.

Parameters:
<SymbName> string

Example: `INST:NAME 'MYAFQ','MYOTHERAFQ','MYSMU'`
Defines the symbolic name of the instruments in the device list.

INSTruments:REMote:CHANnel <Channel#1>,<Channel#2>

Defines the remote channels of the instruments in the device list.

Parameters:
<Channel> LAN | GPIB | USB

Example: `INST:REM:CHAN USB,LAN,GPIB,LAN`
Sets the hardware channels of the first four devices in the list.

INSTruments:REMote:NAME <Name#1>,<Name#2>, ...

Defines the instrument (device) names of the instruments in the device list. The instrument name of an instrument is used to access the instrument over the network.

Parameters:
<Name> string

Example:
```
INST:REM:NAME
'RSAFQ100A1000001','RSAFQ100A1000002',
'RSSMU200A1000025'
```
defines the network names of the three devices in the list.

INSTruments:SCAN <Scan>

Triggers a scan function to search for instruments connected via one of the possible interfaces, LAN, GPIB or USB.

Parameters:
<Scan> ON | OFF | 1 | 0

Example: `INST:SCAN ON`
starts a scan.
```
INST:SCAN?
Response: 1
```
the scan process is still running
```
Response: 0
```
the scan process is completed and the list of available instruments can be queried (`INST:NAME?`)

INSTruments:SELect:ARB <Arb>

Selects one of the instruments available in the network as an ARB instrument

Parameters:

<Arb>	Range: 0 = "Undefined" to Number of available instruments

Example:

```
INST:SCAN ON
```
starts a scan of the network
```
INST:SCAN?
```
Queries whether the scan process is still running
```
Response: 0
```
the scan process is completed, i.e. the list of available instruments can be queried
```
INST:NAME?
```
Queries the list of the available instruments
```
Response: "AFQ100AB10","SMUB10","SMBVB50B55"
INST:SEL:ARB 2
```
Selects the second instrument in the list (in this case the instrument with name SMUB10) as an ARB instrument.
```
INST:SEL:ARB 0
```
Selects an “Undefined” instrument

INSTruments:SELect:VECTor <Vector>

Selects one of the instruments available in the network as a vector signal generator

Parameters:

<Vector>	Range: 0 = "Undefined" to Number of available instruments

Example:

```
INST:SCAN ON
```
starts a scan of the network
```
INST:SCAN?
```
Queries whether the scan process is still running
```
Response: 0
```
the scan process is completed, i.e. the list of available instruments can be queried
```
INST:NAME?
```
Queries the list of the available instruments
```
Response: "AFQ100AB10","SMUB10","SMBVB50B55"
INST:SEL:VEC 2
```
Selects the second instrument in the list (in this case the instrument with name SMUB10) as an ARB instrument.
```
INST:SEL:VEC 0
```
Selects an “Undefined” instrument

INSTruments:TYPE <Type>

Defines the instrument type in the device list, e.g., R&S AFQ, R&S SMU, R&S AMU, etc.

Parameters:

<Type>	AFQB10 | AFQB11 | SMUB9 | SMUB10 | SMUB11 | SMJB9 | SMJB10 | SMJB11 | SMJB50 | SMJB51 | SMATEB9 | SMATEB10 | SMATEB11 | AMUB9 | AMUB10 | AMUB11 | CMW | AFQ100BB12 | AFQ100BB11 | SMBVB10 | SMBVB50 | SMBVB51 | SMBVB10B55 | SMBVB50B55 | SMBVB51B55

Example:

```
INST:TYPE AFQB10,AMUB11,SMUB10
```

defines the device type of one of the three devices in the list.

INSTruments:USB:SERial <Serial#1>,<Serial#2>,...

Defines the serial numbers of the instruments in the device list. The serial number of an instrument is required to access it over USB.

Parameters:

<Serial> string

Example:

```
INST:USB:Serial 100001,100002,100025
```

defines the USB serial number of the three devices in the list.

TRANsmit:AUTO:PATH <Path>

Defines on which path of the instrument the signal is to be generated, if a transmitted waveform file shall be automatically loaded and started on the instrument.

Parameters:

<Path> A | B

Example:

```
TRAN:AUTO:PATH A
```

defines the generation path of the signal.

TRANsmit:AUTO[:STATe] <State>

Defines whether the transmitted waveform file should be automatically loaded and started on the instrument after the waveform file has been transmitted.

Parameters:

<State> 0 | 1 | OFF | ON

Example:

```
TRAN:AUTO:STATe On
```

defines the automatically loading of the waveform.

TRANsmit:COMMent <Comment>

Specifies a comment which will be written into the header of generated waveform file.

Parameters:

<Comment> string

Example:

```
TRAN:COMM 'This is a comment'
```

sets a comment.

TRANsmit:DESTination <Destination>

Defines the transmission destination of the waveform file. There are two options: The Instrument (arbitrary signal generator) or a file on a local drive.

Parameters:

<Destination> INSTrument | FILE

Example:

```
TRAN:DEST INST
```

sets the transmission destination to the instrument.

TRANsmit:DESTination:IFILe <Ifile>

Defines the name of the waveform destination file on the instrument, if `INSTrument` is selected as the transmit destination.

Parameters:

<Ifile> string

Example:

```
TRAN:DEST INST
```

sets the transmission destination to the instrument.

```
TRAN:DEST:IFIL 'D:\3GPP.WV'
```

sets the name of the destination file on the instrument.

TRANsmit:DESTination:LFILe <Lfile>

Defines the name of the waveform destination file on a local hard disc (file copy), if `FILE` is the selected transmit destination.

Parameters:

<Lfile> string

Example:

```
TRAN:DEST FILE
```

sets the transmission destination to file.

```
TRAN:DEST:LFIL 'c:\temp\3gpp.wv'
```

sets the destination of the waveform file to the local PC hard disc.

TRANsmit:SOURce <Source>

Defines the source of the waveform to be transmitted. The waveform file can be generated internally or generated internally and stored locally.

Parameters:

<Source> INTernal | FILE

Example:

```
TRAN:SOUR INT
```

Sets the generation of the waveform file to internal. The waveform is not stored.

TRANsmit:SOURce:LFILe <Lfile>

Defines the name of a source file, that is stored locally and is to be transmitted

Parameters:

<Lfile> string

Example: `TRAN:SOUR:LFIL 'c:\temp\3gpp.wv'`

sets the name of a source file.

TRANsmit:STATe <State>

Starts/stops a transmission. The query informs about the current state of a transmission.

Parameters:

<State> ON | OFF

Example: `TRAN:STAT ON`

starts a transmission.

7.5 SOURce:AWGN Subsystem

The `SOURce:AWGN` subsystem contains the commands for setting the noise generator.

[:SOURce<hw>]:AWGN:BRATe <BRate>

This command sets the bit rate which is used for calculation of bit energy to noise power ratio from carrier/noise ratio for Digital Standard signals

For "Custom Digital Mod" signals, the bit rate which is used for calculation can be queried with this command.

Valid units are bps, kbps and mabps as well as b/s, kb/s and mab/s.

Parameters:

<BRate>	float Range: 400 bps to 250E6 bps Increment: 0.001 *RST: 100000
Example:	`AWGN:BRAT?` queries the bit rate which is used for calculation of the Eb/N0 value from the C/N value.

[:SOURce<hw>]:AWGN:BWIDth <BWidth>

This command sets the system bandwidth. The noise signal at the level which corresponds to the specified carrier/noise ratio is generated in the bandwidth specified here.

This command is available for modes Additive Noise and Noise Only (`SOUR:AWGN:MODE ADD|ONLY`).

Parameters:

<BWidth>	float Range: 1.0E3 to 100.0E6 Increment: 0.1 kHz *RST: 3.84 MHz
Example:	`AWGN:BWID 10 MHz` sets a system bandwidth of 10 MHz.
Options:	B13 and K62

[:SOURce<hw>]:AWGN:BWIDth:NOISe?

Queries the real noise bandwidth.

This command is available for modes Additive Noise and Noise Only (`SOUR:AWGN:MODE ADD|ONLY`).

Return values:

<Noise>	float Range: 0 to 200E6 Increment: 100 *RST: 0
Example:	`AWGN:BWID:NOIS?` queries the noise bandwidth.
Usage:	Query only

[:SOURce<hw>]:AWGN:BWIDth:RATio <Ratio>

This command is available for modes **Additive Noise** and **Noise Only** (`SOUR:AWGN:MODE ADD|ONLY`).

This command sets the ratio of minimum real noise bandwidth to system bandwidth.

The value range of this parameter is automatically adjusted depending on the selected instrument. The maximum ratio minimum noise bandwidth to system bandwidth is determinate by the max sample clock of the selected instrument and is calculated so that the overall bandwidth, i.e. "System BW" x "Minimum Noise"/"System BW Ratio" has to be less or equal to the sampling rate of the selected instrument.

Parameters:

<Ratio>	float	
	Range:	1 to Max
	Increment:	0.1
	*RST:	1

Example:

```
AWGN:BWID:RAT 2
```

sets a minimum noise/system bandwidth ratio of 2.

[:SOURce<hw>]:AWGN:CNRatio <CnRatio>

This command sets the carrier/interferer ratio. The value range depends on the selected AWGN mode (`AWGN:MODE ADD|ONLY|CW`).

This command is available for modes "Additive Noise" and "CW Interferer" (`SOUR:AWGN:MODE ADD|CW`).

Parameters:

<CnRatio>	float	
	Range:	-50.00 to 40.00
	Increment:	0.01
	*RST:	0
	Default unit:	dB

Example:

```
AWGN:CNR 10
```

sets a carrier/noise ratio of 10 dB.

[:SOURce<hw>]:AWGN:ENRatio <EnRatio>

This command sets the ratio of bit energy to noise power density in "Additive Noise" mode.

For **Digital Standard** signals, the bit rate used for calculation of E_b/N_0 value from C/N value can be entered with command `SOUR:AWGN:BRAT`.

For **Custom Digital Mod** signals the bit rate used for calculation of E_b/N_0 value from C/N value is determined by the selected standard (`SOURce:BB:DM:STANdard`) and cannot be changed.

Parameters:

<EnRatio> float

Range: -30 to 30
Increment: 0.001
*RST: 0
Default unit: dB

Example: `AWGN:ENR 10`
sets a ratio of bit energy to noise power density of 10 dB

[:SOURce<hw>]:AWGN:FREQuency:RESult?

This command queries the actual frequency of the sine in the **CW Interferer** mode.

The resulting "CW Frequency Offset" is the correction of the desired value based on generating an integer multiple of periods out of the output sequence length.

Return values:

<Result> float

Range: -40E6 to 40E6
Increment: 0.01
*RST: 0

Example: `AWGN:FREQ:RES?`
queries the actual frequency of the interfering sine

Usage: Query only

[:SOURce<hw>]:AWGN:FREQuency:TARGet <Target>

This command sets the desired frequency of the sine in "CW Interferer" mode (`AWGN:MODE CW`).

The range of this parameter is limited to ± (Symbol Rate + Sample Rate)/4

Parameters:

<Target> float

Range: -40E6 to 40E6
Increment: 0.01
*RST: 0

Example: `AWGN:FREQ:TARG 2kHz`
sets a frequency of 2 kHz for the interfering sine.

[:SOURce<hw>]:AWGN:MODE <Mode>

This command selects the mode for generating the interfering signal.

Parameters:

<Mode> ONLY | ADD | CW

ADD
The AWGN noise signal is added to the baseband signal.

ONLY
The pure AWGN noise signal is modulated to the carrier. The connection to the baseband is interrupted

CW
The sine interfering signal is added to the baseband signal.

*RST: ADD

Example: `AWGN:MODE ONLY`
activates the generation of a pure noise.
`AWGN:STAT ON`
switches on the generation of a pure noise.

[:SOURce<hw>]:AWGN:POWer:CARRier <Carrier>

available for modes "Additive Noise" and "CW Interferer" (`SOUR:AWGN:MODE ADD|CW`)

Parameters:

<Carrier> float

Range: min to 20
Increment: 0.01
*RST: 0
Default unit: dBm

Example: `AWGN:POW:CARR?`
queries the carrier level

[:SOURce<hw>]:AWGN:POWer:MODE <Mode>

This command selects the mode for setting the noise level.

This command is available for mode "Additive Noise" (`SOUR:AWGN:MODE ADD`).

Parameters:

<Mode> CN | SN | EN

CN|SN
available for mode Additive Noise (`SOUR:AWGN:MODE ADD`).
The noise level is set on the basis of the value entered for the carrier/noise or signal/noise ratio (`AWGN:CNR|SNR`).

EN
The noise level is set on the basis of the value entered for the ratio of bit energy to noise power density (`AWGN:ENR`).

*RST: CN|SN

Example: `SOUR:AWGN:POW:MODE CN`
the noise level is set on the basis of the value entered for the carrier/noise ratio (`AWGN:CNR`).

[:SOURce<hw>]:AWGN:POWer:NOISe <Noise>

Available for modes Additive Noise, Noise Only and CW Interferer (`SOUR:AWGN:MODE ADD|ONLY|CW`).

This command sets or queries the noise level in the system bandwidth depending on the selected modes:

- "Additive Noise" and "CW Interferer" mode (`SOUR:AWGN:MODE ADD|CW`)
  The command queries the noise/interferer level which is derived from the entered S/N value.
- "Noise Only" mode
  The command sets the noise level.

Parameters:

<Noise> float

Default unit: dBm

Example: `SOUR:AWGN:POW:NOIS?`
queries the noise level in the system bandwidth.
Response: `10`
the noise level in the system bandwidth is 10 dBm.

[:SOURce<hw>]:AWGN:POWer:NOISe:TOTal?

Available for modes Additive Noise and Noise Only (`SOUR:AWGN:MODE ADD|ONLY`)

This command queries the noise level in the total bandwidth.

Return values:

<Total> float

Range:	-145 to 20
Increment:	0.01
*RST:	-30

Example: `SOUR:AWGN:POW:NOIS:TOT?`
queries the noise level in the total bandwidth.
Response: `15`
the noise level in the total bandwidth is 15 dBm.

Usage: Query only

[:SOURce<hw>]:AWGN:POWer:SUM?

Available for modes "Additive Noise" and "CW Interferer" (`SOUR:AWGN:MODE ADD|CW`)

This command queries the overall level of the noise signal plus useful signal.

Return values:

<Sum> float

Range:	-145 to 20
Increment:	0.01
*RST:	0

Example: `SOUR:AWGN:POW:SUM?`
queries the overall level of the noise signal plus useful signal.

Usage: Query only

[:SOURce<hw>]:AWGN:POWer:SUM:PEP?

Available for modes Additive Noise and CW Interferer (`SOUR:AWGN:MODE ADD|CW`)

This command queries the peak envelope power of the overall signal comprised of noise signal plus useful signal.

Return values:

<Pep> float

Range:	-145 to 20
Increment:	0.01
*RST:	0

Example: `SOUR:AWGN:POW:SUM:PEP?`
queries the peak envelope power of the overall signal.

Usage: Query only

[:SOURce<hw>]:AWGN:SLENgth <SLength>

This command is available for mode Noise Only (`SOUR:AWGN:MODE ONLY`).

Sets the sequence length of the signal in samples.

The sequence length depends on the configured instrument. If the configured instrument is changed, the value range of this parameter is changed as well.

Parameters:

<SLength> integer

Range:	128 to 1073741824
Increment:	1

Example: `AWGN:MODE ONLY`
activates the generation of a pure noise.
`AWGN:STAT ON`
switches on the generation of a pure noise.
`AWGN:SLEN 1024`
sets the sequence legth.

[:SOURce<hw>]:AWGN:SRATe <SRate>

This command sets the bit rate which is used for calculation of bit energy to noise power ratio from carrier/noise ratio for Digital Standard signals. For Custom Digital Mod signals, the bit rate which is used for calculation can be queried with this command.

Parameters:

<SRate>	float Range: 400 to 250E6 bps Increment: 0.001 *RST: 40E6 Default unit: Valid units are bps, kbps and Mbps.

Example: `AWGN:BRAT?`
queries the bit rate which is used for calculation of the Eb/N0 value from the C/N value.

[:SOURce<hw>]:AWGN:STATe <State>

This command activates or deactivates the white noise (AWGN = Averaged White Gaussian Noise). The noise signal is either superimposed on the baseband signal ("Additive Noise" mode (`SOUR:AWGN:MODE ADD`)) or is output as a pure noise signal ("Noise Only" mode (`SOUR:AWGN:MODE ONLY`)). In addition, a sine with adjustable frequency offset to the baseband signal can be generated as an RFI signal ("CW Interferer" mode; `SOUR:AWGN:MODE CW`).

Parameters:

<State>	0 | 1 | OFF | ON *RST: 0

Example: `AWGN:STAT ON`
activates the RFI signal generator.

7.6 SOURce:BB:ARB:MCAR Subsystem

The `MCARrrier` subsystem contains the commands for setting the Multi Carrier Waveform Generator.

CARRier<ch>

The numerical suffix under `CARRier` distinguish between the carriers. The value range is 0 .. 31.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier:COUNt <Count>

The command sets the number of carriers in the ARB multi carrier waveform.

The total bandwidth (**Number of carriers - 1**) ***Carrier spacing** is 80 MHz.

The number of carriers entered therefore defines the maximum carrier spacing (`:BB:ARB:MCARrier:CARRier:SPACing`) and the carrier spacing is reduced if the total bandwidth is not respected when entering the number of carriers.

Parameters:

<Count> integer

Range:	1 to 512
Increment:	1
*RST:	1

Example: `BB:ARB:MCAR:CARR:COUN 10`
sets 10 carriers for the multi carrier waveform.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier:MODE <mode>

The command sets the carrier frequency mode.

Parameters:

<mode> EQUidistant | ARBitrary

EQUidistant
Sets an equidistant carrrier spacing. The carrier frequency in the carrier table is not configurable.

ARBitrary
Enables you to specify the carrier frequency in the carrier table. Carrier spacing is irrelevant.

*RST: EQUidistant

Example:

```
BB:ARB:MCAR:CARR:MODE EQU
```

sets an equidistant carrier spacing. The carrier frequency can not be set.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier:SPACing <Spacing>

The command sets the frequency spacing between adjacent carriers of the multi carrier waveform. The carriers are generated symmetrically around the RF carrier. The maximum carrier spacing is limited to **Carrier spacing = Total baseband bandwidth /(Number of carriers - 1).**

The total baseband bandwidth is 80 MHz.

Note: In order to avoid wrap-around problems, the effective "Carrier Spacing" might be slightly modified. The "Carrier Spacing" is rounded in that way that the carrier closest to the center RF frequency shows no phase jump assuming that the carrier is unmodulated.

- For odd number of carriers:
  RoundedCarrierSpacing=1/OutputSignalDuration* round(CarrierSpacing * OutputSignalDuration);
- For even number of carriers:
  RoundedCarrierSpacing=2/OutputSignalDuration*round(0.5 *CarrierSpacing * OutputSignalDuration);

Parameters:

<Spacing> float

Range: 0.0 to 120E6
Increment: 0.01
*RST: 0
Default unit: Hz

Example:

```
BB:ARB:MCAR:CARR:SPAC 10 MHz
```

sets a carrier spacing of 10 MHz.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:CONFlict?

Queries carrier conflicts. A conflict arises when the carriers overlap.

Return values:

<conflict> 0 | 1 | OFF | ON

*RST: 0

Example: `BB:ARB:MCAR:CARR2:CONF?`
queries the multi carrier conflict state.
Response: `0` no conflict has occurred.

Usage: Query only

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:DELay <Delay>

The command sets the start delay of the selected carrier.

Parameters:

<Delay> float

Range: 0 to 1
Increment: 1E-9
*RST: 0
Default unit: s

Example: `BB:ARB:MCAR:CARR15:DEL 5us`
sets a start delay of 50 us for carrier 15.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:FILE <File>

The command selects the file with I/Q data to be modulated onto the selected carrier.

Parameters:

<File> <file name>

Example: `BB:ARB:MCAR:CARR15:FILE 'D:/IQ_wcdma'`
selects file `IQ_wcdma`. The data of the file is modulated onto carrier 15.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:FREQuency <Frequency>

Sets or indicates the carrier frequency, depending on the selected carrier frequency mode.

The carrier frequnecy can be set in "Arbitrary Carrier frequency" mode. For "Equidistant Carrier Spacing", the carrier spacing is determined automatically.

Parameters:

<Frequency> integer

Range: -40.0E6 to 40.0E6
Increment: 1
*RST: 0

Example: `BB:ARB:MCAR:CARR:MODE ARB`
activates ARB multi carrier mode.
`BB:ARB:MCAR:CARR:FREQ 5.0`
sets 5.0 MHz carrier frequency.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:PHASe <Phase>

The command sets the start phase of the selected carrier.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:ARB:MCARrier:CFACtor:MODE OFF`).

Parameters:

<Phase> float

Range: 0 to 359.99
Increment: 0.01
*RST: 0
Default unit: DEG

Example: `BB:ARB:MCAR:CARR15:PHAS 90 DEG`
sets a start phase of 90° for carrier 15.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:POWer <Power>

The command sets the gain of the selected carrier.

Parameters:

<Power> float

Range: -80 to 0
Increment: 0.01
*RST: 0
Default unit: dB

Example: `BB:ARB:MCAR:CARR15:POW -50 dB`
sets the power of carrier 15 to -50 dB.

[:SOURce<hw>]:BB:ARB:MCARrier:CARRier<ch>:STATe <State>

The command switches the selected carrier on or off.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: ON

Example: `BB:ARB:MCAR:CARR15:STAT ON`
switches carrier 15 on.

[:SOURce<hw>]:BB:ARB:MCARrier:CFACtor:MODE <Mode>

The command sets the mode for optimizing the crest factor by calculating the carrier phases.

Parameters:

<Mode> OFF | MIN | MAX

OFF
There is no automatic setting for minimizing or maximizing the crest factor. The Phase setting (command `BB:ARB:MCAR:CARR:PHAS`) is in use.

MIN
The crest factor is minimized by internally calculating optimized carrier phases. The Phase setting (command `BB:ARB:MCAR:CARR:PHAS`) is invalid.

MAX
The crest factor is maximized by internally calculating optimized carrier phases. The Phase setting (command `BB:ARB:MCAR:CARR:PHAS`) is invalid.

*RST: OFF

Example:

`BB:ARB:MCAR:CFAC:MODE OFF`
switches off automatic crest factor optimization. The setting `SOUR:BB:ARB:MCAR:CARR:PHAS` has an effect.

[:SOURce<hw>]:BB:ARB:MCARrier:CLIPping:CFACtor <CFactor>

Sets the value of the desired crest factor, if baseband clipping is enabled (`BB:ARB:MCAR:CLIP:STAT ON`).

A Target Crest Factor above the crest factor of the unclipped multicarrier signal has no effect.

Parameters:

<CFactor> float

Range:	-50 to 50
Increment:	0.01
*RST:	50
Default unit:	dB

Example:

`BB:ARB:MCAR:CLIP:STAT ON`
enables clipping.
`BB:ARB:MCAR:CLIP:CFAC 37 dB`
sets the target crest factor.

[:SOURce<hw>]:BB:ARB:MCARrier:CLIPping:CUToff <Cutoff>

Sets the cut off frequency of the final lowpass filter, if baseband clipping is enabled (`BB:ARB:MCAR:CLIP:STAT ON`).

When the cut off frequency is set as half of the output sample rate, a final lowpass filter improves the spectrum of the clipped multicarrier signal, but may also increase the resulting crest factor.

Parameters:

<Cutoff> float

Range:	0 to 50E6
Increment:	0.01
*RST:	50E6
Default unit:	MHz

Example:

```
BB:ARB:MCAR:CLIP:STAT ON
```

enables clipping

```
BB:ARB:MCAR:CLIP:CUT 50 MHz
```

sets the cut off frequency of the filter.

[:SOURce<hw>]:BB:ARB:MCARrier:CLIPping[:STATe] <State>

Switches baseband clipping on and off.

Clipping reduces the peak power of the resulting multi carrier signal according to the value set with the command `BB:ARB:MCAR:CLIP:CFAC`.

The resulting clipped peak power is defined by sum of the the RMS level of the unclipped multi carrier signal and the input parameter Target Crest Factor. Note that clipping reduces also the RMS level. Hence the resulting crest factor of the clipped signal is slightly above the Target Crest Factor. In order to get the unclipped parts of the clipped multicarrier signal matched with the unclipped multicarrier signal, the RF output power should be reduced by the difference between resulting crest factor and Target Crest Factor.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example:

```
BB:ARB:MCAR:CLIP:STAT ON
```

enables clipping

```
BB:ARB:MCAR:CLIP:CFAC 37 dB
```

sets the target crest factor.

[:SOURce<hw>]:BB:ARB:MCARrier:CLOad <Cload>

This command creates a multi carrier waveform using the current entries of the carrier table.

This multi carrier waveform is saved with the file name specified with command `SOUR:BB:ARB:MCAR:OFIL`. The file extension is `*.wv`. Digital standard "ARB" is activated, the new multi carrier waveform is loaded and is output in accordance to the trigger settings.

Parameters:

<Cload> <file name>

Example:

`MMEM:CDIR 'D:/user/waveform'`

sets the default directory to `D:/user/waveform`.

`BB:ARB:MCAR:OFIL 'mcar1_2'`

defines the file name `mcar1_2.wv` for the multi carrier waveform.

`BB:ARB:MCAR:CLO`

creates multi carrier waveform `mcar1_2.wv`.

The new multi carrier waveform is loaded and digital standard "ARB" is activated.

Usage: Event

[:SOURce<hw>]:BB:ARB:MCARrier:CLOCk?

The command queries the resulting sample rate at which the multi carrier waveform is output by the arbitrary waveform generator. The output clock rate depends on the number of carriers, carrier spacing and input sample rate of the leftmost or rightmost carriers.

Return values:

<Clock> float

Range: 400 to Max
Increment: 1E-3

Example:

`BB:ARB:MCAR:CLOC?`

queries the ARB multi carrier output clock rate.

Usage: Query only

[:SOURce<hw>]:BB:ARB:MCARrier:CREate

This command creates a multi carrier waveform using the current settings of the carrier table. The multi carrier waveform is saved into the file defined with command `SOUR:BB:ARB:MCAR:OFIL`. The file extension is `*.wv`.

Example:

`MMEM:CDIR 'D:/user/waveform'`

sets the default directory to `D:/user/waveform`.

`BB:ARB:MCAR:OFIL 'multi_wv1'`

defines the file name `multi_wv1.wv` for the multi carrier waveform.

`BB:ARB:MCAR:CRE`

creates multi carrier waveform `multi_wv1.wv`.

Usage: Event

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:DELay:STEP <Step>

The command sets the step width by which the start delays of the carriers in the defined carrier range will be incremented.

Parameters:

<Step>	float Range: -1 to 1 Increment: 1E-9 *RST: 0 Default unit: s

Example:

`BB:ARB:MCAR:EDIT:CARR:DEL 5 us`

sets a start delay of 5 us for the carriers in the carrier range.

`BB:ARB:MCAR:EDIT:CARR:DEL:STEP 1 us`

the start delay is incremented by 1us for each carrier, i.e. the first carrier has a start delay of 5us, the second a start delay of 6 us, etc.

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:DELay[:STARt] <Start>

The command sets the start delay for the individual carriers in the defined carrier range. If the command `:BB:ARB:MCAR:EDIT:CARR:DEL:STEP` is used to define a step width, the delay entered here applies only to the starting carrier. The delays of the remaining carriers are stepped up or down by the delay value specified in the `:BB:ARB:MCAR:EDIT:CARR:DEL:STEP` command.

Parameters:

<Start>	float Range: 0 to 1 Increment: 1E-9 *RST: 0 Default unit: s

Example:

`BB:ARB:MCAR:EDIT:CARR:DEL 5us`

sets a start delay of 5 us for the carriers in the carrier range.

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:EXECute

The command adopts the settings for the carrier range which has been defined using the `:BB:ARB:MCAR:EDIT:CARR:...` commands.

Example:

`BB:ARB:MCAR:EDIT:CARR:STAR 4`

the carrier range starts at carrier 2.

`BB:ARB:MCAR:EDIT:CARR:STOP 20`

the carrier range stops at carrier 20.

`BB:ARB:MCAR:EDIT:CARR:STAT ON`

sets all the carriers in the carrier range (2 to 20) to ON.

`BB:ARB:MCAR:EDIT:CARR:EXEC`

transfers the assistant settings for carrier 2 to 20 into the carrier table.

Usage: Event

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:FILE <File>

Parameters:

<File> string

Example:
```
BB:ARB:MCAR:EDIT:CARR:FILE 'D:/IQ_wcdma'
```
selects input file `IQ_wcdma`. The data of the file are modulated onto the carriers in the defined carrier range.

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:PHASe:STEP <Step>

The command sets the step width by which the start phases of the carriers in the defined carrier range will be incremented.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:ARB:MCARrier:CFACtor:MODE OFF`).

Parameters:

<Step> float

Range: -359.99 to 359.99
Increment: 0.01
*RST: 0
Default unit: DEG

Example:
```
BB:ARB:MCAR:EDIT:CARR:PHAS 90 DEG
```
sets a start phase of 90° for the carriers in the carrier range.
```
BB:ARB:MCAR:EDIT:CARR:PHAS:STEP 1 DEG
```
the start phase is incremented by 1° for each carrier, i.e. the first carrier has a start phase of 90°, the second a start phase of 91°, etc.

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:PHASe[:STARt] <Start>

The command sets the start phase for the individual carriers in the defined carrier range. If the command `:BB:ARB:MCAR:EDIT:CARR:PHAS:STEP` is used to define a step width, the phase entered here applies only to the starting carrier. The phases of the remaining carriers are stepped up or down by the phase value specified in the `:BB:ARB:MCAR:EDIT:CARR:PHAS:STEP` command.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:ARB:MCARrier:CFACtor:MODE OFF`).

Parameters:

<Start> float

Range: 0 to 359.99
Increment: 0.01
*RST: 0
Default unit: DEG

Example:
```
BB:ARB:MCAR:EDIT:CARR:PHAS 90 DEG
```
sets a start phase of 90° for the carriers in the carrier range.

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:POWer:STEP <Step>

The command sets the step width by which the starting power of the carriers in the defined carrier range will be incremented.

Parameters:

<Step> float

Range:	-80 to 80
Increment:	0.01
*RST:	0
Default unit:	dB

Example:

```
BB:ARB:MCAR:EDIT:CARR:POW -80dB
```

sets a power of -80 dB for the carriers in the carrier range.

```
BB:ARB:MCAR:EDIT:CARR:POW:STEP 1 dB
```

the power is incremented by 1dB for each carrier, i.e. the first carrier has -80dB, the second -79dB, etc.

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:POWer[:STARt] <Start>

The command sets the power for the individual carriers in the defined carrier range. If the command `:BB:ARB:MCAR:EDIT:CARR:POW:STEP` is used to define a step width, the power entered here applies only to the starting carrier. The power of the remaining carriers is stepped up or down by the power specified in the `:BB:ARB:MCAR:EDIT:CARR:POW:STEP` command.

Parameters:

<Start> float

Range:	-80 to 0
Increment:	0.01
*RST:	0
Default unit:	dB

Example:

```
BB:ARB:MCAR:EDIT:CARR:POW -50 dB
```

sets the power of the carriers in the carrier range to -50 dB.

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:STARt <Start>
[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:STOP <Stop>

The command selects the last carrier in the carrier range to which the settings with the `:BB:ARB:MCAR:EDIT:CARR:..` commands shall apply.

Parameters:

<Stop> integer

Range:	0 to 511
Increment:	1
*RST:	0

Example:

```
BB:ARB:MCAR:EDIT:CARR:STOP 4
```

the carrier range stops at carrier 4.

[:SOURce<hw>]:BB:ARB:MCARrier:EDIT:CARRier:STATe <State>

The command switches all the carriers in the selected carrier range on or off.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 1

Example: `BB:ARB:MCAR:EDIT:CARR:STAT ON`
sets all the carriers in the carrier range to ON.

[:SOURce<hw>]:BB:ARB:MCARrier:OFILe <OFile>

This command defines the output file name for the multi carrier waveform. This file name is used when a waveform is calculated (command `SOUR:BB:ARB:MCAR:CLOad` or `SOUR:BB:ARB:MCAR:CREate`). The file extension is `*.wv`.

Parameters:

<OFile> string

Example: `MMEM:CDIR 'D:/user/waveform'`
sets the default directory to D:/user/waveform.
`BB:ARB:MCAR:OFIL 'mcar1_2'`
defines the file name `mcar1_2.wv` for the multi carrier waveform file

[:SOURce<hw>]:BB:ARB:MCARrier:PRESet

The command sets all ARB multi carrier parameters to their default values.

Example: `BB:ARB:MCAR:PRES`
resets the ARB multi carrier parameters to default values.

Usage: Event

[:SOURce<hw>]:BB:ARB:MCARrier:SAMPles?

The command queries the resulting file size. The file size is returned in samples.

Return values:

<Samples> integer

Range: 0 to INT_MAX
Increment: 1
*RST: 0

Example: `BB:ARB:MCAR:SAMP?`
queries the file size of the currently calculated multi carrier waveform.

Usage: Query only

[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:CATalog?

This command queries the available settings files in the specified default directory. The settings files are used to set the ARB multi carrier submenu. Only files with the file extension `*.arb_multcarr` will be listed.

Return values:

<Catalog>	string

Example:	`MMEM:CDIR 'D:/user/waveform'`
	sets the default directory to D:/user/waveform.
	`BB:ARB:MCAR:SETT:CAT?`
	reads out all the settings files in the default directory.
	`Response: mcar1, mcar2`
	the directory contains the configuration files `mcar1.arb_multcarr` and `mcar2.arb_multcarr`.

Usage:	Query only

[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:LOAD <Filename>

The command loads the settings file. If a settings file with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.arb_multcarr` will be loaded or created.

Setting parameters:

<Filename>	string
Example:	`BB:ARB:MCAR:SETT:LOAD 'D:/user/waveform/new'` creates settings file `new.arb_multcarr`.
Usage:	Setting only

[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:STORe <Filename>

The command stores the current settings of submenu "Multi Carrier" in a file in the specified directory. The file extension may be omitted, the files are stored with the file extension `*.arb_multcarr`.

Setting parameters:

<Filename>	string
Example:	`BB:ARB:MCAR:SETT:STOR 'D:/user/waveform/mcarr2'` stores settings file `mcarr2.arb_multcarr` in the default directory.
Usage:	Setting only

[:SOURce<hw>]:BB:ARB:MCARrier:SETTing:STORe:FAST <Fast>

Determines whether the instrument performs an absolute or a differential storing of the settings.

Enable this function to accelerate the saving process by saving only the settings with values different to the default ones.

Note: This function is not affected by the "Preset" function.

Parameters:

<Fast> 0 | 1 | OFF | ON

*RST: 1

[:SOURce<hw>]:BB:ARB:MCARrier:STATe <State>

The command enables/disables multi carrier generation.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: OFF

Example:

```
BB:ARB:MCAR:STAT ON
```

enables multi carrier generation.

[:SOURce<hw>]:BB:ARB:MCARrier:TIME:MODE <Mode>

The command selects the mode for calculating the resulting signal period of the multi carrier waveform. The resulting period is always calculated for all carriers in the carrier table irrespective of their state (ON/OFF).

Parameters:

<Mode> USER | LONG | SHORt

LONG
The resulting signal period is defined by the longest I/Q file in the carrier table. Shorter I/Q files are periodically repeated.

SHORt
The resulting signal period is defined by the shortest I/Q file in the carrier table. Only the first part of longer I/Q files is used.

USER
The signal period can be set with command `SOUR:BB:ARB:MCARr:TIME`. Shorter I/Q files are repeated periodically, and only the first part of longer I/Q files is used.

*RST: LONG

Example:

```
BB:ARB:MCAR:TIME:MODE LONG
```

selects signal period mode long

7.7 SOURce:BB:ARB:WSEG Subsystem

The `WSEGment` subsystem contains the commands for setting the Multi Segment Waveform Generator.

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:BLANk:APPend <SampCount>, <Frequency>

Adds a blank segment to the multi segment file.

Setting parameters:

<SampCount> float

Specifies the number of samples.

Range: 512 to 1E7
Increment: 1
*RST: 1000

<Frequency> float

Determines the clock rate.

Range: 400 Hz to Max (depends on instrument type)
Increment: 0.001
*RST: 1E8

Example:

```
SOUR:BB:ARB:WSEG:CONF:SEL "MSegFile"
```

selects a multi segment file.

```
SOUR:BB:ARB:WSEG:CONF:BLAN:APP 1000,100000000
```

adds a blank segment with 1000 samples and 100 MHz clock rate to the selected multi segment file

Usage: Setting only

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:CATalog?

This command queries the available configuration files in the specified default directory. The configuration files are used to create multi segment waveform files.

Return values:

<Catalog>	string

Example:

```
MMEM:CDIR 'D:/user/waveform'
```

sets the default directory to D:/use/waveform.

```
BB:ARB:WSEG:CONF:CAT?
```

reads out all the configuration files in the default directory.

```
Response: mulit1, multi2
```

the directory contains the configuration files `multi1.inf_mswv` and `multi2.inf_mswv`.

Usage: Query only

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:CLOCk <Clock>

This command defines the clock rate used for multi segment waveform output in case of Clock Mode "User" (`:BB:ARB:WSEG:CONF:CLOCk:MODE USER`).

Parameters:

<Clock> float

Increment: 1E-3
*RST: max SampleRate

Example:

```
BB:ARB:WSEG:CONF:CLOC:MODE USER
```

selects Clock Mode User.

```
BB:ARB:WSEG:CONF:CLOC 50MHz
```

defines a clock rate of 50 MHz.

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:CLOCk:MODE <Mode>

This command selects the clock rate mode for the multi segment waveform.

Parameters:

<Mode> UNCHanged | HIGHest | USER

UNCHanged
The segments are output with the clock rate defined in the waveform file. Extended Trigger Mode "Next Segment Seamless" (:BB:ARB:TRIG:SMOD NSEam) can only be selected if all segments have the same clock rate. Extended Trigger Mode "Next Segment" (:BB:ARB:TRIG:SMOD NEXT) can only be selected if trigger mode "Internal" is selected (:BB:ARB:TRIG:SOUR INT).

HIGHest
The segments are output at the highest available clock rate.

USER
The segments are output with the clock rate defined with command `:BB:ARB:WSEG:CONF:CLOC`

*RST: UNCHanged

Example: `BB:ARB:WSEG:CONF:CLOC:MODE UNCH`
selects clock mode unchanged. The segments are output with the clock rate defined in the waveform file.

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:COMMent <Comment>

This command enters a comment for the configuration file. The configuration file must be specified with command `:BB:ARB:WSEG:CONF:SEL`.

Parameters:

<Comment> string

Example: `BB:ARB:WSEG:CONF:COMM <3gpp_up>`
enters comment "3gpp_up".

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:DELete <Filename>

This command deletes the configuration file. The configuration files are used to create multi segment waveform files.

Setting parameters:

<Filename> string

Example: `MMEM:CDIR 'D:/user/waveform'`
sets the default directory to D:/user/waveform.
`BB:ARB:WSEG:CONF:DEL 'multi1'`
deletes configuration file `multi1.inf_mswv`.

Usage: Setting only

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:LEVel[:MODE] <Mode>

This command selects the level mode for the multi segment waveform.

Parameters:

<Mode> UNCHanged | ERMS

UNCHanged
The segments are output exactly as defined in the files.

ERMS
The segments are output so that all segments have the same rms value.

*RST: UNCHanged

Example: `BB:ARB:WSEG:CONF:LEV:MODE UNCH`
selects level mode unchanged. The segments are output as defined in the waveform file.

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:MARKer:ESEGment <Mode>

Enables/disables the generation of an additional marker restart signal in the output waveform file.

If additional marker generation is enabled, the existing marker signals in the individual segment waveform files are not considered.

Parameters:

<Mode> OFF | MRK1 | MRK2 | MRK3 | MRK4

OFF
No additional marker is generated.

MRK1|MRK2|MRK3|MRK4
Generates a restart marker signal at the beginning of each segment for the corresponding marker.
The segment begin is defined by the low-high slope of the marker. This applies for switching between two segments as well as in case of segment replay.

*RST: OFF

Example:

```
BB:ARB:WSEG:CONF:SEL 'ConfComm'
BB:ARB:WSEG:CONF:MARK:ESEG MRK3
```

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:MARKer:FSEGment <Mode>

Enables/disables the generation of an additional marker restart signal in the output waveform file.

If additional marker generation is enabled, the existing marker signals in the individual segment waveform files are not considered.

Parameters:

<Mode> OFF | MRK1 | MRK2 | MRK3 | MRK4

OFF
No additional marker is generated.

MRK1|MRK2|MRK3|MRK4
Generates a restart marker signal at the beginning of the first segment for the corresponding marker.
Use this setting to generate a restart marker for the complete sequence.

*RST: OFF

Example:

```
BB:ARB:WSEG:CONF:SEL 'ConfComm'
BB:ARB:WSEG:CONF:MARK:FSEG MRK3
```

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:MARKer:MODE <Mode>

Defines the way the marker information within the separate segments is processed.

Parameters:

<Mode> IGNore | TAKE

IGNore
The marker information carried in the individual segment waveform files is not considered.

TAKE
The output waveform file contains the marker information as configured in the individual waveform files.

*RST: TAKE

Example:

```
BB:ARB:WSEG:CONF:SEL 'ConfComm'
BB:ARB:WSEG:CONF:MARK:MODE TAKE
```

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:OFILe <OFile>

This command defines the file name for the multi segment waveform. This file name is stored in the configuration file and is used when a waveform is created (command `:BB:ARB:WSEG:CLOad` or `:BB:ARB:WSEG:CREate`). The file extension is `*.wv`.

Parameters:

<OFile> string

Example:

`MMEM:CDIR 'D:/user/waveform'`
sets the default directory to D:/user/waveform.
`BB:ARB:WSEG:CONF:SEL 'multi1'`
creates the configuration file `multi1.inf_mswv` in default directory.
`BB:ARB:WSEG:CONF:OFIL 'mseg1_2'`
defines the file name `mseg1_2.wv` for the multi segment waveform file created using configuration file `multi1.inf_mswv`.

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:SEGMent:APPend <Waveform>

This command appends the specified waveform to the configuration file.

Setting parameters:

<Waveform> string

Example:

`MMEM:CDIR 'D:/user/waveform'`
sets the default directory to D:/user/waveform.
`BB:ARB:WSEG:CONF:SEL 'new'`
creates the configuration file `new.inf_mswv` in the default directory.
`BB:ARB:WSEG:CONF:SEGM:APP 'arb1'`
appends waveform `arb1.wv` to configuration file `new`. Waveform `arb1` will be the first segment of a multi segment waveform created with configuration file `new`.

Usage: Setting only

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:SEGMent:CATalog?

This command queries the segments of the currently selected configuration file.

Return values:
<Catalog> string

Example:
`MMEM:CDIR 'D:/user/waveform'`
sets the default directory to D:/user/waveform.
`BB:ARB:WSEG:CONF:SEL 'multi_sin'`
selects the configuration file multi_sin.inf_mswv .
`BB:ARB:WSEG:CONF:SEGM:CAT?`
queries the segments of the selected configuration file.
`Response: arb4, arb2`
The configuration file includes the segments `arb4.wv` and `arb2.wv`.

Usage: Query only

[:SOURce<hw>]:BB:ARB:WSEGment:CONFigure:SELect <Filename>

The command selects the configuration file in the default directory A path can also be specified, in which case the files in the specified directory are selected. If a configuration file with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.inf_mswv` will be selected or created.

Parameters:
<Filename> string

Example:
`MMEM:CDIR 'D:/user/waveform'`
sets the default directory to D:/user/waveform.
`BB:ARB:WSEG:CONF:SEL 'new'`
create configuration file `new.inf_mswv` in the default directory.

[:SOURce<hw>]:BB:ARB:WSEGment:CREate <Filename>

This command creates a multi segment waveform using the current settings of the specified configuration file. The multi segment waveform is saved into the file defined in the configuration file. The file extension is `*.wv`.

Setting parameters:
<Filename> string

Example:

`MMEM:CDIR 'D:/user/waveform'`
sets the default directory to D:/user/waveform.
`BB:ARB:WSEG:CONF:SEL 'new'`
creates the configuration file `new.inf_mswv` in the default directory.
`BB:ARB:WSEG:CONF:SEGM:APP 'arb1.wv'`
'includes waveform `arb1.wv` as segment 1 in the configuration file. The waveform must be available in the default directory.
`BB:ARB:WSEG:CONF:OFIL 'multi_wv1'`
defines the file name `multi_wv1.wv` for the multi segment waveform.
`BB:ARB:WSEG:CRE 'new'`
creates multi segment waveform `multi_wv1.wv` using the settings of the configuration file `new.inf_mswv`.

Usage: Setting only

[:SOURce<hw>]:BB:ARB:WSEGment:NEXT <Next>

This command selects the segment to be output.

Parameters:

<Next> integer

Range:	0 to 1023
Increment:	1
*RST:	0

Example:

`MMEM:CDIR 'D:/user/waveform'`
sets the default directory to root/user/waveform.
`BB:ARB:WSEG:NEXT 2`
selects segment 2 to be output.
`BB:ARB:TRIG:SMOD NEXT`
selects extended trigger mode next, i.e. the segment specified with command `:BB:ARB:WSEG:NEXT` will be output.
`BB:ARB:SEQ AUTO`
selects trigger mode Auto.
`BB:ARB:WAV:SEL 'multi_wv1'`
loads multi segment waveform `multi_wv1.wv`. Generation of segment 2 starts.
`BB:ARB:WSEG:NEXT 3`
switched at once to output of segment 3.

[:SOURce<hw>]:BB:ARB:WSEGment:NEXT:EXECute

Triggers manually switchover to the subsequent segment in the mutli segment file. A manual trigger can be executed only when an internal next segment source (`BB:ARB:WSEG:NEXT:SOUR INT`) has been selected.

To perform a switchover to any segment within the multi segment file, select the next segment with the command `BB:ARB:WSEG:NEXT`.

This command is disabled, if a sequencing play list is enabled.

Example: `BB:ARB:WSEG:NEXT:SOUR INT`
selects internal next segment source.
`BB:ARB:WSEG:NEXT 2`
selects segment 2 to be output.
`BB:ARB:WSEG:NEXT:EXEC`
executes a switchover to the next segment.

Usage: Event

[:SOURce<hw>]:BB:ARB:WSEGment:SEQuence:APPend <State>, <Segment>, <Count>, <Next>

Appends a new segment to the selected sequencing play list.

Setting parameters:

<State> ON | OFF

activates/deactivates the appended segment

*RST: ON

<Segment> integer

indicates the number of the segment as in the multi segment waveform file

Range: 0 to SegmentCount - 1
Increment: 1

<Count> integer

defines how many times this segment is repeated

Range: 1 to 65535
Increment: 1

<Next> NEXT | BLANk | ENDLess | SEG0 | SEG1 | ... | SEG31 | 0...maxSegment

determines the action after completing the current segment, like for instance which segment is processed after the processing of the current one is finished.

Example: `BB:ARB:WSEG:SEQ:SEL 'D:/user/waveform/play_list_1'`
selects the sequencing list `play_list_1.wvs`.
`BB:ARB:WSEG:SEQ:APP ON,3,15,BLANK`
appends the segment number 3 as a new segment to the sequencing list; this segment is activated and will be repeated 15 times and followed by a blank segment.

Usage: Setting only

[:SOURce<hw>]:BB:ARB:WSEGment:SEQuence:SELect <Filename>

Selects the sequencing list (files with extension `*.wvs`)

Parameters:

<Filename> string

Example: `BB:ARB:WSEG:SEQ:SEL 'D:/user/waveform/play_list_1'`
selects the sequencing list `play_list_1.wvs`.

[:SOURce<hw>]:BB:ARB:WSEGment:STATe <State>

This command enables/disables multi segment generation.

Parameters:

<State> 0 | 1 | OFF | ON
*RST: 1

Example: `BB:ARB:WSEG:STAT ON`
enables multi segment generation.

7.8 SOURce:BB:DM Subsystem

The commands in the `SOURce:BB:DM` subsystem are described in two sections, separated into configuring digital modulation and lists for digital modulation.

7.8.1 DM General Remote-Control Commands

The following section contains the commands for generating the digital modulation signal.

7.8.1.1 Common Settings

[:SOURce<hw>]:BB:DM:PATTern <Pattern>

The command selects the data pattern for the internal data when PATTern is selected as the data source. The maximum length is 64 bits.

Parameters:

<Pattern> integer

Range: #B0,1 to #B111...1,64
*RST: #B0,1

Example: `BB:DM:SOUR PATT`
selects Pattern as the data source for digital modulation.
`BB:DM:PATT #B01110111010101010,17`
generates the user-defined sequence of 0/1 data.

[:SOURce<hw>]:BB:DM:PRBS[:LENGth] <Length>

The command defines the length of the pseudo-random sequence in accordance with the following equation:

Length = (2^`Length`) - 1

Parameters:

<Length> 9 | 11 | 15 | 16 | 20 | 21 | 23

*RST: 9

Example: `BB:DM:SOUR PRBS`
The internal pseudo-random generator is used as the data source.
`BB:DM:PRBS 9`
An internal pseudo-random sequence of 511 bits will be generated.

[:SOURce<hw>]:BB:DM:PRESet

The command calls the default settings for digital modulation.

Example: `BB:DM:PRES`
calls the default settings for DM.

Usage: Event

[:SOURce<hw>]:BB:DM:SOURce <Source>

The command selects the data source.

Parameters:

<Source> ZERO | ONE | PRBS | PATTern | DLISt

ZERO
An internally generated 0 data sequence is used.

ONE
An internally generated 1 data sequence is used.

PRBS
The pseudo-random sequence generator is used as the data source.The length of the random sequence is defined with the aid of command `SOURce:BB:DM:PRBS`.

PATTern
Internally generated data with a fixed pattern is used. The data pattern is dafined using command `SOURce:BB:DM:PATTern`.

DLISt
Data from the selected data list is used.

*RST: PRBS

Example: `BB:DM:SOUR DLIS`
the internal data generator is used.
`BB:DM:DLIS:SEL 'test'`
the data list `test.dm_iqd` is used.

[:SOURce<hw>]:BB:DM:SRATe <SRate>

The command sets the symbol rate. The value range is dependent on the selected modulation type. On changing to an FSK modulation type, excessively high values are automatically limited to the maximum value that can be set for FSK (see data sheet). The symbol rate can be entered in Hz/kHz/MHz or Symb/s / kSymb/s and MSymb/s.

When a standard is selected (`DM:STANdard`), the symbol rate is automatically set to the appropriate default value.

The value range is dependent on the selected modulation type (:`BB:DM:FORM`). When a standard is selected (`:BB:DM:STAN`), the symbol rate is set to the default value.

Parameters:

<SRate> float

Range: 400Hz to 15MHz (FSK) / 25 MHz
Increment: 0.001Hz
*RST: 270833kHz
Default unit: Hz

Example: `BB:DM:SRAT 10 MHz`
sets a symbol rate of 10 MHz.

[:SOURce<hw>]:BB:DM:STANdard <Standard>

The command selects the standard. After selection, modulation parameters "Modulation Type" (`:BB:DM:FORMat`), "Symbol Rate" (`:BB:DM:SRATe`), "Filter" (`:BB:DM:FILTer:TYPE` and `:BB:DM:FILTer:PARameter:...`) and "Coding" (`:BB:DM:CODing`) are automatically set in accordance with the standard. The USER parameter cannot be set. A query returns this value if a user-defined Custom Dig Mod setting was loaded or if one of the associated settings was changed subsequent to the selection of a standard. The user defined settings are stored and loaded with commands `:BB:DM:STAN:ULIS:...` (see chapter 7.8.2, "DM Lists", on page 244).

Parameters:

<Standard> USER | BLUetooth | DECT | ETC | GSM | GSMEdge | NADC | PDC | PHS | TETRa | W3GPp | TDSCdma | CFORward | CREVerse | WORLdspace | TFTS | APCOPH1C4fm | APCOPH1CQpsk | APCOPH2HCpm | APCOPH2HDQpsk | APCOPH2HD8PSKW | APCOPH2HD8PSKN | APCOPH1Lsm | APCOPH1Wcqpsk

*RST: GSM

Example:

`BB:DM:STAN DECT`

selects digital modulation according to the DECT standard.

[:SOURce<hw>]:BB:DM:STATe <State>

The command enables or disables digital modulation. Switching on digital modulation turns off all the other digital standards.

Enabling digital modulation (`BB:DM:STAT ON`) turns off all the other standards.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: OFF

Example:

`BB:DM:STAT ON`

switches digital modulation on.

[:SOURce<hw>]:BB:DM:SLENgth <Slength>

Sets the sequence length of the signal in symbols. The signal is calculated in advance, saved as waveform file, and output in the selected arbitrary waveform generator.

Note: Sequence Length * Oversampling must not exceed the maximum number of samples in the arbitrary waveform generator.

Parameters:

<Slength> float

Range: 1.0 to max

*RST: 10000

Example:

`BB:DM:SLEN 500`

sets the sequence length to 500 symbols.

[:SOURce<hw>]:BB:DM:WAVeform:CREate <Filename>

This command creates a waveform using the current settings of the "Digital Modulation" menu. The file name is entered with the command. The file is stored with the predefined file extension `*.wv`. The file name and the directory it is stored in are user-definable.

Setting parameters:
<Filename> string

Example: `MMEM:CDIR 'd:/user/waveform'`
sets the default directory to `d:/user/waveform`.
`BB:DM:WAV:CRE 'dm_1`
creates the waveform file `dm_1.wv` in the default directory.

Usage: Setting only

7.8.1.2 Marker Settings

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:MODE <Mode>

The command defines the signal for the selected marker output.

Parameters:

<Mode> CLISt | PULSe | PATTern | RATio | TRIGger

CLISt
A marker signal that is defined in the selected control list is generated.

PULSe
A pulsed marker signal is generated. The pulse frequency (= symbol rate/divider) is defined with the `SOUR:BB:DM:TRIG:OUTP:PULSe:DIVider` command and can be queried with the `SOUR:BB:DM:TRIG:OUTP:PULSe:FREQuency?` command.

PATTern
A marker signal is generated with the aid of a user-definable bit pattern. The bit pattern is entered with the aid of command `SOURce:BB:DM:TRIGger:OUTPut:PATTern`. The bit pattern is a maximum of 32 bits long.

RATio
A regular marker signal corresponding to the Time Off / Time On specifications in the commands `SOURce:BB:DM:TRIGger:OUTPut:OFFT` and `SOURce:BB:DM:TRIGger:OUTPut:ONT` is generated.

*RST: RATio

Example: `BB:DM:TRIG:OUTP2:MODE PULS`
selects the pulsed marker signal on output MARKER 2.

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:OFFTime <OffTime>

The command sets the number of symbols in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:DM:TRIGger:OUTPut:MODE RATio` on the marker outputs is OFF.

Parameters:

<OffTime> integer

Range: 1 to 16777215
Increment: 1
*RST: 1
Default unit: symbol

Example: `BB:DM:TRIG:OUTP2:OFFT 20`
sets an OFF time of 20 symbols for marker signal 2.

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:ONTime <OnTime>

The command sets the number of symbols in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:DM:TRIGger:OUTPut:MODE RATio` on the marker outputs is ON.

Parameters:

<OnTime>	float Range: 1 to 16777215 *RST: 1 Default unit: symbol

Example:
```
BB:DM:TRIG:OUTP2:ONT 20
```
sets an ON time of 20 symbols for marker 2

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PATTern <Pattern>

The command defines the bit pattern used to generate the marker signal in the setting `SOURce:BB:DM:TRIGger:OUTPut:MODE PATTern`. 0 is marker off, 1 is marker on.

Parameters:

<Pattern>	integer Range: #B0,1 to #B111...1,32 *RST: #B,1

Example:
```
BB:DM:TRIG:OUTP2:PATT #B000000011111111,15
```
sets a bit pattern.
```
BB:DM:TRIG:OUTP2:MODE PATT
```
activates the marker signal according to a bit pattern on output MARKER 2.

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PULSe:DIVider <Divider>

The command sets the divider for Pulse marker mode (`SOUR:BB:DM:TRIGr:OUTP:MODE PULSe`). The resulting pulse frequency is derived by dividing the symbol rate by the divider.

Parameters:

<Divider>	integer Range: 2 to 1024 Increment: 1 *RST: 2

Example:
```
BB:DM:TRIG:OUTP2:PULS:DIV 2
```
sets the divider to 2 for the marker signal on output MARKER 2.
```
BB:DM:TRIG:OUTP2:FREQ?
```
queries the resulting pulse frequency of the marker signal.
```
Response: 66 000
```
the resulting pulse frequency is 66 kHz.

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PULSe:FREQuency?

The command queries the pulse frequency of the pulsed marker signal in the setting `SOURce:BB:DM:TRIGger:OUTPut:MODE PULSe`. The pulse frequency is derived by dividing the symbol rate by the divider.

Return values:

<Frequency> float

Example:

```
BB:DM:TRIG:OUTP2:PULS:DIV 2
```

sets the divider marker signal on output MARKER 2 to the value 2.

```
BB:DM:TRIG:OUTP2:MODE PULS
```

enables the pulsed marker signal.

```
BB:DM:TRIG:OUTP2:PULS:FREQ?
```

queries the pulse frequency of the marker signal.

```
Response: 33 000
```

the resulting pulse frequency is

Usage: Query only

7.8.1.3 Filter Settings

[:SOURce<hw>]:BB:DM:FILTer:ILENgth <ILength>

The command sets the impulse length (number of filter tabs).

Parameters:

<ILength> integer

Range: 1 to 128
Increment: 1
*RST: 10

Example:

```
BB:DM:FILT:ILEN 10
```

sets the number of filter tabs to 10.

[:SOURce<hw>]:BB:DM:FILTer:ILENgth:AUTO <Auto>

The command activates/deactivates the impulse length state. If activated, the most sensible parameter values are selected. The value depends on the coherence check.

Parameters:

<Auto>	0 | 1 | OFF | ON *RST: 1

Example: `BB:DM:FILT:ILEN:AUTO ON`
The most sensible parameters are selected automatically.

[:SOURce<hw>]:BB:DM:FILTer:OSAMpling <OSampling>

The command sets the upsampling factor.

Parameters:

<OSampling>	integer Range: 1 to 32 Increment: 1 *RST: 32

Example: `BB:DM:FILT:OSAM 32`
sets the upsampling factor to 32.

[:SOURce<hw>]:BB:DM:FILTer:OSAMpling:AUTO <Auto>

The command activates/deactivates the upsampling factor state. If activated, the most sensible parameter values are selected. The value depends on the coherence check. If deactivated, the values can be changed manually.

Parameters:

<Auto>	0 | 1 | OFF | ON *RST: 1

Example: `BB:DM:FILT:OSAM:AUTO ON`
The most sensible parameters are selected automatically.

[:SOURce<hw>]:BB:DM:FILTer:PARameter:COSine[:ROLLoff] <Cosine>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:GAUSs <Gauss>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:LPASs <LPass>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:LPASSEVM <LPassEvm>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:PGAuss <PGauss>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:RCOSine <RCosine>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:SPHase <SPhase>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25 <Apco25>

The command sets the roll-off factor for filter type APCO25.

On selecting filter APCO25, the filter parameter is set to the default value.

Parameters:

<Apco25>	float Range: 0.05 to 0.99 Increment: 0.01 *RST: 0.2

<Cosine> float

Range: 0.05 to 1.0
Increment: 0.01
*RST: 0.35

<Gauss> float

Range: 0.15 to 100000
Increment: 0.01
*RST: 0.3

<LPass> float

Range: 0.05 to 2
Increment: 0.01
*RST: 0.5

<LPassEvm> float

Range: 0.05 to 2
Increment: 0.01
*RST: 0.5

<PGauss> float

Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.3

<RCosine> float

Range: 0.05 to 1.0
Increment: 0.01
*RST: 0.35

<SPhase> float

Range: 0.15 to 2.5
Increment: 0.01
*RST: 2

Example:

```
BB:DM:FILT:PAR:APCO25 0.2
```

sets the roll-off factor to 0.2 for filter type APCO25.

[:SOURce<hw>]:BB:DM:FILTer:PARameter:COSine:BANDwidth <FiltParm>

Sets the bandwidth of the cosine filter.

Parameters:

<FiltParm> float

Range: 400 to 25E6
Increment: 1E-3
*RST: 270833.333

Example:

```
:BB:DM:FILT:TYPE COS
:BB:DM:FILT:PAR:COS:BAND 7200
```

[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25Lsm:GAUSs <FiltParm>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25Lsm:LOWPass <FiltParm>

Sets the cut off frequency for the lowpass/ gauss filter.

Parameters:

<FiltParm> float

Range:	400 to 25E6
Increment:	1E-3
*RST:	270833.333

Example:

```
:BB:DM:FILT:TYPE APCO25Lsm
:BB:DM:FILT:PAR:APCO25L:LOWP 5166.667
:BB:DM:FILT:PAR:APCO25L:GAUS 1700
```

[:SOURce<hw>]:BB:DM:FILTer:TYPE <Type>

The command selects the filter type.

When a standard is selected (`:BB:DM:STAN`), the filter type and filter parameter are set to the default value.

Parameters:

<Type> RCOSine | COSine | GAUSs | LGAuss | CONE | COF705 | COEQualizer | COFequalizer | C2K3x | APCO25 | SPHase | RECTangle | USER | PGAuss | LPASs | DIRac | ENPShape | EWPShape | LTEFilter | LPASSEVM | APCO25Hcpm | APCO25Lsm

*RST: GAUSs

Example:

```
BB:DM:FILT:TYPE COS
```

selects the Cosine filter type.

7.8.1.4 Modulation Settings

[:SOURce<hw>]:BB:DM:AQPSk:ANGLe <Angle>

For AQPSK modulation, sets the angle alpha between the point (0,0) and the I axis.

Parameters:

<Angle> float

Range: 0 to 180
Increment: 0.01
*RST: 0
Default unit: Deg

Example:

`BB:DM:FORM AQPS`
selects modulation type AQPSK.
`BB:DM:AQPS:ANGL 45`

[:SOURce<hw>]:BB:DM:ASK:DEPTh <Depth>

The command sets the ASK modulation depth when modulation type ASK is selected.

Parameters:

<Depth> float

Range: 0 to 100
Increment: 0.1PCT
*RST: 100
Default unit: PCT

Example:

`BB:DM:FORM ASK`
selects the ASK modulation type.
`BB:DM:ASK:DEPT 50 PCT`
sets a modulation depth of 50 percent.

[:SOURce<hw>]:BB:DM:CODing <Coding>

The command selects the modulation coding.

If the modulation type selected (`:BB:DM:FORM`) is not possible with the coding that has been set, it is automatically set to OFF (`:BB:DM:COD OFF`).

When a standard is selected (`:BB:DM:STAN`), the coding is set to the default value.

Parameters:

<Coding> OFF | DIFF | DPHS | DGRay | GRAY | GSM | NADC | PDC | PHS | TETRa | APCO25 | PWT | TFTS | INMarsat | VDL | EDGE | APCO25FSK | ICO | CDMA2000 | WCDMA | APCO258PSK

DPHS
Phase Difference

DGRay
Difference + Gray

*RST: INTernal

Example:

`BB:DM:COD GRAY`
selects GRAY coding. This coding is valid for all modulation types.

[:SOURce<hw>]:BB:DM:FORMat <Format>

The command selects the modulation type.

If the coding that is set (`:BB:DM:COD`) is not possible with the modulation type selected, it is automatically set to OFF (`:BB:DM:COD OFF`).

When a standard is selected (`:DM:STAN`), the modulation type is set to the default value.

Parameters:

<Format> ASK | BPSK | P2DBpsk | QPSK | QPSK45 | OQPSk | P4QPsk | P4DQpsk | PSK8 | P8D8psk | P8EDge | QAM16 | QAM32 | QAM64 | QAM256 | QAM1024 | MSK | FSK2 | FSK4 | USER | FSKVar | QAM128 | QEDGe | QAM16EDge | QAM32EDge | AQPSk

*RST: MSK

Example:

`BB:DM:FORM QPSK`

selects modulation type QPSK.

[:SOURce<hw>]:BB:DM:FSK:DEViation <Deviation>

The command sets the frequency deviation when FSK modulation is selected. The range of values depends on the symbol rate that is set (see data sheet) and the maximum deviation is 10 MHz.

If the symbol rate that is set exceeds the maximum possible value for the chosen frequency deviation, it is suitably adapted (`:BB:DM:SRAT`).

Parameters:

<Deviation> float

Range: 1 to 40E6
Increment: 0.5
*RST: 135416.5

Example:

`BB:DM:FORM FSK`

selects FSK modulation.

`BB:DM:FSK:DEV 10 MHz`

sets the frequency deviation to 10 MHz.

[:SOURce<hw>]:BB:DM:FSK:VARiable:SYMBol<ch>:DEViation <Deviation>

The command sets the deviation of the selected symbol for variable FSK modulation mode.

Suffix:

<ch> 0 .. 16

The number of symbols (and therefore the suffix range) depends on the selected FSK modulation type.

Parameters:

<Deviation> float

The range of values depends on the symbol rate that is set (see data sheet) and the maximum deviation is 10 MHz.

Range: -40E6 to 40E6
Increment: 0.5
Default unit: Hz

Example:

```
BB:DM:FORM FSKV
```
selects Variable FSK modulation.
```
BB:DM:FSK:VAR:TYPE FSK16
```
selects 16FSK modulation.
```
BB:DM:FSK:VAR:SYMB0:DEV 135000
```
sets the frequency deviation of the least significant symbol to 135 kHz.

[:SOURce<hw>]:BB:DM:FSK:VARiable:TYPE <Type>

The command selects the modulation type for Variable FSK.

Parameters:

<Type> FSK4 | FSK8 | FSK16

*RST: FSK4

Example:

```
BB:DM:FORM FSKV
```
selects Variable FSK modulation.
```
BB:DM:FSK:VAR:TYPE FSK16
```
selects 16FSK modulation.

7.8.1.5 Power Ramp

[:SOURce<hw>]:BB:DM:PRAMp:ATTenuation <Attenuation>

The command sets the level attenuation for signal ranges that are flagged with level attribute **attenuated** by the LEV_ATT control signal.

Parameters:

<Attenuation> float

Range: 0 to 50
Increment: 0.1
*RST: 15
Default unit: dB

Example:

```
BB:DM:PRAM:ATT 15 dB
```

sets a level attenuation of 15 dB.

[:SOURce<hw>]:BB:DM:PRAMp:FDELay <FDelay>

The command sets the delay in the falling edge. A positive value gives rise to a delay and a negative value causes an advance. The setting is expressed in symbols.

Parameters:

<FDelay> float

Range:	-4 to 4
Increment:	0.01
*RST:	0
Default unit:	symbol

Example:

```
BB:DM:PRAM:FDEL 1
```

The falling edge starts 1 symbol later.

[:SOURce<hw>]:BB:DM:PRAMp:RDELay <RDelay>

The command sets the delay in the rising edge. A positive value gives rise to a delay and a negative value causes an advance. The setting is expressed in symbols.

Parameters:

<RDelay> float

Range:	-4 to 4
Increment:	0.01
*RST:	0
Default unit:	symbol

Example:

```
BB:DM:PRAM:RDEL 1
```

The rising edge starts 1 symbol later.

[:SOURce<hw>]:BB:DM:PRAMp:SHAPe <Shape>

The command sets the edge shape of the ramp envelope.

Parameters:

<Shape> LINear | COSine

*RST: COSine

Example:

```
BB:DM:PRAM:SHAP COS
```

selects a cosine-shaped rise and fall for the transmitted power edge.

[:SOURce<hw>]:BB:DM:PRAMp:TIME <Time>

The command sets the power ramping rise time and fall time for a burst.

Parameters:

<Time> float

Range: 0.25 to 16
Increment: 0.01
*RST: 1
Default unit: symbol

Example: `BB:DM:PRAM:TIME 2`
sets a time of 2 symbols for the edges to rise and fall.

[:SOURce<hw>]:BB:DM:PRAMp[:STATe] <State>

The command enables or disables power ramping.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `BB:DM:PRAM:STAT ON`
switches power ramping on.

7.8.2 DM Lists

The following section brings together the commands for defining and managing the data lists and control lists for digital modulation.

Lists are stored as files with specific file extensions in a user-definable directory (see table 7-1). The directory which applies to the commands that follow is defined using the command `MMEMory:CDIR`. Files in this directory can be accessed by quoting the file name. The path and extension are not needed.

Table 7-1: Overview of the used list types

List type	Content	File extension
Data List	Digital modulation data	`*.dm_iqd`
Control List	Digital modulation control data	`*.dm_iqc`
User Standards	User settings of digital modulation	`*.dm_stu`
Usr Filter	Usr filter settings	`*.vaf`
User Mapping	User mapping settings	`*.vam`

It is not possible to use other file extensions with the commands. Attempting to do so will cause an error message. If the file extension is changed in any other way (e.g. by directly accessing the file system) the lists are no longer recognized and therefore invalid.

Since the format of control lists refer to the input, the `CLISt` commands always create a binary control list, even if ASCII format is specified. Note therefore, if you want to create a `*.wv` file with R&S WinIQSIM2 using a control list, you have to use the input control list in ASCII format. See the example in chapter 7.2.3, "Creating an ASCII C-List Manually", on page 185 for information on how to create a C-list in ASCII format.

List of Commands

[:SOURce<hw>]:BB:DM:CLISt:CATalog?
[:SOURce<hw>]:BB:DM:DLISt:CATalog?

[:SOURce<hw>]:BB:DM:FLISt:CATalog?
[:SOURce<hw>]:BB:DM:MLISt:CATalog?

Queries the user mapping lists present in the default directory. The default directory is set using command `MMEM:CDIRectory`. When the names of the files are returned they are separated by commas.

The command only reads out files with the `*.vam` extension (see also table 7-1).

Return values:

<Catalog> string

Example:

`BB:DM:FORM USER`
selects the User modulation type.
`MMEM:CDIR 'D:/user/mapping_List'`
sets the default directory for the user mapping lists.
`BB:DM:MLIS:CAT?`
queries the user mapping lists in the directory.

Usage: Query only

[:SOURce<hw>]:BB:DM:CLISt:FREE?
[:SOURce<hw>]:BB:DM:DLISt:FREE?
[:SOURce<hw>]:BB:DM:FLISt:FREE?
[:SOURce<hw>]:BB:DM:MLISt:FREE?

Queries the user modulation mapping list free memory.

Return values:

<Free> integer

Range:	0 to INT_MAX
Increment:	1
*RST:	0

Example:

`BB:DM:FORM USER`
selects the User modulation type.
`BB:DM:FLIS:FREE?`
queries the free memory.

Usage: Query only

[:SOURce<hw>]:BB:DM:CLISt:COPY <Filename>
[:SOURce<hw>]:BB:DM:DLISt:COPY <Filename>

The command copies the selected data list into the data list specified by <list name>. If a data list with the specified name already exists, it is overwritten. If it does not yet exist, it is created.

The source file has to be available in the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can be specified, in which case the source file is copied into the file <list name> in the specified directory. The file extension may be omitted.

Only files with the file extension `*.dm_iqc` will be copied (see also table 7-1).

Setting parameters:
<Filename> string

Example: `BB:DM:DLIS 'd_list1'`
selects data list `d_list1`.
`BB:DM:DLIS:COPY 'd_list2'`
copies the content of data list `d_list1` into data list `d_list2`.
Any existing content in data list `d_list2` is overwritten.

Usage: Setting only

[:SOURce<hw>]:BB:DM:CLISt:TAG?
[:SOURce<hw>]:BB:DM:DLISt:TAG?

The command queries the content of the specified tag in the selected file.

Return values:
<Tag> <control list>,<tag name>

Example: `BB:DM:DLIS:TAG 'D_list1','date'`
queries the Date tag in control list `D_list1`.
Response:`10.10.2008`
the data list was created on 10.10.2008.

Usage: Query only

[:SOURce<hw>]:BB:DM:CLISt:DATA <Data>

The command sends the data to the currently selected control list. If the list already contains data, it is overwritten. This command only writes data into the data section of the file.

The values for the control signals are sent, arranged in a 8-bit value as follows:

Signal	Order	Decimal value of bits
Marker 1 Marker 2	LSBit	1 2
Burst =	LSBit	16
LevAtt1	LSBit	32
CWMod	LSBit	64
Hop	MSBit	128

The data can also be sent as a binary block, each binary block being a 2-byte value in which the 16 bits represent the binary values (16-bit unsigned integer, 2 bytes, LSByte first). When binary data transmission is in use, command `:SYSTem:COMMunicate:GPIB:LTERminator EOI` should be used to set the termination character mode to ‘EOI control data message only’ so that a random LF in the data sequence is not interpreted as End, thereby prematurely terminating the data transmission. The command `...LTER STANdard` resets the mode.

For query purposes, the command `:FORMat ASCii | PACKed` can be used to switch between the formats. The byte sequence is defined in the IEC bus standard as 'most significant byte first'

Note: C-Lists are created in binary format. However, if you need the C-List in ASCII format, e.g. for creating a waveform file with R&S WinIQSIM2, see the example in chapter 7.2.3, "Creating an ASCII C-List Manually", on page 185.

`*RST` has no effect on data lists. This command is without query.

Setting parameters:

<Data> string

Example: `BB:DM:CLIS:SEL 'c_list1'`
selects the control list.
`BB:DM:CLIS:DATA 0,0,0,0,8,8,8,0,0,0,0...`
enters the control values in list c_list1. In the example, only ramps for marker 4 are set.

Usage: Setting only

[:SOURce<hw>]:BB:DM:CLISt:DELete <Filename>

The command deletes the specified control list from the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are deleted. The file extension may be omitted. Only files with the file extension `*.dm_iqc` will be deleted.

Setting parameters:

<Filename> <list name>

Example: `BB:DM:CLIS:DEL 'c_list3'`
deletes control list c_list3.

Usage: Setting only

[:SOURce<hw>]:BB:DM:CLISt:POINts?

The command queries the number of lines (2 bytes) in the currently selected list.

Return values:

<Points> integer

Range:	0 to INT_MAX
Increment:	1
*RST:	0

Example: `BB:DM:CLIS:SEL "c_list1"`
selects control list c_list1.
`BB:DM:CLIS:POIN?`
queries the number of lines in the control list.
Response: `20`
the control list consists of 20 lines.

Usage: Query only

[:SOURce<hw>]:BB:DM:CLISt:SELect <Filename>

The command selects the control list in the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are selected. If a control list with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.dm_iqc` will be selected or created.

Parameters:
<Filename> <list name>

Example: `BB:DM:CLIS:SEL 'c_list1'`
selects control list `c_list1`.

[:SOURce<hw>]:BB:DM:DLISt:DATA <Data>

The **Setting** command sends the bit data to the data list selected with the aid of `:BB:DM:DLISt:SELect`. Any existing content in the data list is overwritten. This command only writes data into the data section of the file.

This command sends the bit data to the selected data list, which is overwritten. The data can also be sent as block data in binary or PACKed format, each byte being interpreted as 8 data bits. In this case, command `:SYSTem:COMMunicate:GPIB:LTERminator EOI` should be used to set the termination character mode to ‘EOI control data message only’ so that a random LF in the data sequence is not interpreted as End, thereby prematurely terminating the data transmission. The command `...LTER STAN` resets the mode.

The **query** reads out the data part of the data list. If the query is expanded by using the two parameters <start> and <length>, the list is read out in smaller sections. Start and Length are expressed in bits. Without the parameters the total length is always read out starting from address 1. The command `:FORMat ASCii | PACKed` can be used to select the data format. The byte sequence is defined in the IEC bus standard (read/write most significant byte first). `*RST` has no effect on data lists.

Parameters:
<Data> integer

Example: `BB:DM:DLIS:SEL 'dlist1'`
selects data list dlist1. If the file does not yet exist, it is created.
`BB:DM:DLIS:DATA 1,1,1,0,0,0,1,1,0,1...`
sends the specified data to file dlist1. Any data already present is overwritten.

Example: `BB:DM:DLIS:SEL 'dlist1'`
selects data list dlist1.
`FORM ASCI`
selects ASCII data transmission format.
`BB:DM:DLIS:DATA? 2048,1024`
queries the data starting at bit 2048 for 1024 bits.

[:SOURce<hw>]:BB:DM:DLISt:DATA:APPend <Bits>

The command appends the bit data onto the end of the existing data in the selected data list. This means that existing content in the data list is not overwritten. By this means very long data lists can be built up piecemeal. The data format is as specified in command `SOURce:BB:DM:DLISt:DATA`.

The command cannot be used with an empty data list, such as one that has just been created, for example. In this case the command `SOURce:BB:DM:DLISt:DATA` must first be used to enter modulation data in the list.

`*RST` has no effect on data lists.

Setting parameters:

<Bits>	0 | 1 {,0 | 1 } | block data

Example:	`BB:DM:DLIS:SEL ' d_list2'`
	selects data list `d_list2`.
	`FORM ASC`
	selects ASCII data transmission format.
	`BB:DM:DLIS:DATA:APP 1,1,1,0,0,0,1,1,0,1...`
	adds the specified numeric data to the existing data in data list `d_list2`.

Usage:	Setting only

[:SOURce<hw>]:BB:DM:DLISt:DELete <Filename>

The command deletes the specified data list. from the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are deleted. The file extension may be omitted. Only files with the file extension `*.dm_iqd` will be deleted.

Setting parameters:

<Filename>	string
Example:	`BB:DM:DLIS:DEL ' d_list2'` deletes data list d_list2.
Usage:	Setting only

[:SOURce<hw>]:BB:DM:DLISt:POINts <Points>

The command defines the number of bits in the selected data list to be utilized. When a list is being filled with block data, this data is only ever sent in multiples of 8 bits. However the exact number of bits to be exploited can be set to a different figure. The superfluous bits in the list are then ignored.

Parameters:

<Points>	integer Range: 0 to INT_MAX Increment: 1 *RST: 0

Example: `BB:DM:DLIS:POIN 234`
defines the number of bits in the data list to be utilized as 234 bits. If the list was filled with block data, at least the last 6 bits will be ignored.

[:SOURce<hw>]:BB:DM:DLISt:SELect <Select>

The command selects the data list in the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are selected. If a data list with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.dm_iqd` will be selected or created.

The modulation data in this data list is used when the data source is selected using the command `SOURce:BB:DM:SOURce DLISt`.

Setting parameters:
<Select> <list name>

Example: `BB:DM:DLIS:SEL 'd_list2'`
selects data list `d_list2`.

Usage: Setting only

[:SOURce<hw>]:BB:DM:FLISt:DELete <Filename>

Deletes the secified user filter file. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.vaf`*.vaf will be deleted.

Setting parameters:
<Filename> string

Example: `BB:DM:FILT:TYPE USER`
selects the User filter type.
`MMEM:CDIR 'D:/user/Filter_List'`
sets the default directory for the user-defined filters.
`BB:DM:FLIS:DEL user_filter3`
deletes the user-defined filter.

Usage: Setting only

[:SOURce<hw>]:BB:DM:FLISt:POINts?

Queries the user filter list length.

Return values:
<Points> integer

Range: 0 to INT_MAX
Increment: 1
*RST: 0

Example: `BB:DM:FILT:TYPE USER`
selects the User filter type.
`BB:DM:FLIS:POIN?`
queries the list length.

Usage: Query only

[:SOURce<hw>]:BB:DM:FLISt:SELect <Filename>

The command selects the user-defined filter (`*.vaf`).

The directory applicable to the following command is defined with the command `MMEM:CDIRectory`. To access the files in this directory, only the file name is required, without the path and the file extension.

Parameters:
<Filename> string

Example: `BB:DM:FILT:TYPE USER`
selects the User filter type.
`MMEM:CDIR 'D:/user/Filter_List'`
sets the default directory for the user-defined filters.
`BB:DM:FLIS:SEL user_filter3`
selects the user-defined filter.

[:SOURce<hw>]:BB:DM:MLISt:DELete <Filename>

Deletes the secified user mapping file. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.vam` will be deleted.

Setting parameters:
<Filename> string

Example: `BB:DM:FORM USER`
selects the User modulation type.
`MMEM:CDIR 'D:/user/mapping_List'`
sets the default directory for the user mapping lists.
`BB:DM:MLIS:DEL user_mapping3`
deletes the user mapping list.

Usage: Setting only

[:SOURce<hw>]:BB:DM:MLISt:POINts?

Queries the user modulation mapping list length.

Return values:

<Points> integer

Range: 0 to INT_MAX
Increment: 1
*RST: 0

Example: `BB:DM:FORM USER`
selects the User modulation type.
`BB:DM:MLIS:POIN?`
queries the list length.

Usage: Query only

[:SOURce<hw>]:BB:DM:MLISt:SELect <Filename>

The command selects the user mapping list in the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are selected. If a user mapping list with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.vam` will be selected or created.

Parameters:

<Filename> string

Example: `BB:DM:MLIS:SEL 'c_list1'`
selects the user mapping list `c_list1`.

[:SOURce<hw>]:BB:DM:SETTing:CATalog?

This command reads out the files with Custom Digital Modulation settings in the default directory. The default directory is set using command `MMEM:CDIRectory`. Only files with the file extension `*.dm` will be listed.

Return values:

<Catalog> string

Example: `MMEM:CDIR 'D:/user/dig_mod'`
sets the default directory.
`BB:DM:SETT:CAT?`
reads out all the files with Custom Digital Modulation settings in the default directory.
Response: `'DM_1'`
the file `DM_1` with Custom Digital Modulation settings is available.

Usage: Query only

[:SOURce<hw>]:BB:DM:SETTing:DELete <Filename>

This command deletes the selected file with Custom Digital Modulation settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.dm` will be deleted.

Setting parameters:

<Filename> string

Example: `BB:DM:STAN:ULIS:DEL 'DM_1'`
deletes file `DM_1`.

Usage: Setting only

[:SOURce<hw>]:BB:DM:SETTing:LOAD <Filename>

This command loads the selected file with Custom Digital Modulation settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.dm` will be loaded.

Setting parameters:

<Filename> string

Example: `BB:DM:STAN:ULIS:LOAD 'DM_1'`
loads file `DM_1`.

Usage: Setting only

[:SOURce<hw>]:BB:DM:SETTing:STORe <Filename>

This command stores the current Custom Digital Modulation settings into the selected file. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. Only the file name has to be entered. User Standards are stored as files with the specific file extensions `*.dm`.

Setting parameters:

<Filename> string

Example: `BB:DM:STAN:ULIS:STOR 'DM_QAM'`
stores the current Custom Digital Modulation settings into file `DM_QAM`.

Usage: Setting only

[:SOURce<hw>]:BB:DM:SETTing:STORe:FAST <Fast>

Determines whether the instrument performs an absolute or a differential storing of the settings.

Enable this function to accelerate the saving process by saving only the settings with values different to the default ones.

Note: This function is not affected by the "Preset" function.

Parameters:

<Fast>	0 | 1 | OFF | ON *RST: 1

[:SOURce<hw>]:BB:DM:STANdard:ULISt:CATalog?

This command reads out the files with Digital Standard settings in the default directory. The default directory is set using command `MMEM:CDIRectory`. Only files with the file extension `*.dm_stu` will be listed.

Return values:

<Catalog> string

Example:

```
MMEM:CDIR 'D:/user/dig_mod'
```

sets the default directory.

```
BB:DM:STAN:ULIS:CAT?
```

reads out all the files with Digital Standard settings in the default directory.

Response: `'DM_QAM'`

the file `DM_QAM` with Digital Standard settings is available.

Usage: Query only

[:SOURce<hw>]:BB:DM:STANdard:ULISt:DELete <Filename>

This command deletes the selected file with Digital Standard settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.dm_stu` will be deleted.

Setting parameters:

<Filename> string

Example:

```
BB:DM:STAN:ULIS:DEL 'DM_QAM'
```

deletes file 'DM_QAM'.

Usage: Setting only

[:SOURce<hw>]:BB:DM:STANdard:ULISt:LOAD <Filename>

This command loads the selected file with Digital Standard settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.dm_stu` will be loaded.

Setting parameters:

<Filename> string

Example:

```
BB:DM:STAN:ULIS:LOAD 'DM_QAM'
```

loads file 'DM_QAM'.

Usage: Setting only

[:SOURce<hw>]:BB:DM:STANdard:ULISt:STORe <Filename>

This command stores the current Digital Standard settings into the selected file. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. Only the file name has to be entered. User Standards are stored as files with the specific file extensions `*.dm_stu`.

Setting parameters:
<Filename> string

Example: `BB:DM:STAN:ULIS:STOR 'DM_QAM'`
stores the current Digital Standard settings into file 'DM_QAM'.

Usage: Setting only

7.9 SOURce:BB:IMPort Subsystem

The IMPort subsystem contains the commands for importing IQ signal data files via TCP/IP into R&S WinIQSIM2.

7.9.1 General Commands

[:SOURce<hw>]:BB:IMPort:PRESet

The command calls the default settings for IQ data import.

Example: `BB:IMP:PRES`
calls the default settings for IQ data import.

Usage: Event

[:SOURce<hw>]:BB:IMPort:SERVer:ID?

The command queries the connected import server. If no import server is connected, **not connected** is displayed. Seraching for a connected server is triggered with "Update Server Connection".

Return values:

<ID> string

Example: `BB:IMP:SERV?`
queries the connected import server.

Usage: Query only

[:SOURce<hw>]:BB:IMPort:SERVer:LOCal[:STATe] <State>

The command acivates or deactivates the local host as the server from where the IQ data streams will be dowloaded. If activated, the field for entering a remote host is filled with the entry "localhost" automatically and cannot be edited.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: ON

Example: `BB:IMP:SERV:LOC:STAT ON`
sets the localhost as the connected server.

[:SOURce<hw>]:BB:IMPort:SERVer:NAME <Name>

The command enters the name (IP address) of the connected server. To enter the name in this field, the check box "Use Local Server" has to be unchecked.

Parameters:

<Name> string

*RST: localhost

Example: `BB:IMP:SERV:NAME 12.34.45.78`
enters the IP address of the connected server.

[:SOURce<hw>]:BB:IMPort:SERVer:PORT <Port>

The command enters the port number for the server connection.

Parameters:

<Port> float

Range: 0 to 65536
*RST: 1000

Example: `BB:IMP:SERV:PORT 1000`
sets port 1000 for the server connection.

[:SOURce<hw>]:BB:IMPort:SERVer:TTOut <Ttout>

The command sets the duration after which the transmission is aborted in case of a server timeout.

Parameters:

<Ttout> float

Range: 10 to 10000
*RST: 20

Example: `BB:IMP:SERV:TTO 20`
sets the transmission timeout to 20 seconds.

[:SOURce<hw>]:BB:IMPort:SERVer:UPDate

The command udates the TCP/IP parameter and tries to establish a connection to the server.

Example: `BB:IMP:SERV:UPD`
updates the server connection.

Usage: Event

[:SOURce<hw>]:BB:IMPort:SETTing:CATalog <Catalog>

This command reads out the files with import settings in the default directory. The default directory is set using command `MMEM:CDIRectory`. Only files with the file extension `*.import` will be listed.

Parameters:

<Catalog> string

Example: `MMEM:CDIR 'D:/user/waveform'`
sets the default directory to D:/user/waveform.
`BB:IMP:SETT:CAT?`
reads out all the files with import settings in the default directory.
Response: 'import_1', 'import_2'
the files `import_1` and `import_2` with import settings are available.

[:SOURce<hw>]:BB:IMPort:SETTing:DELete <Delete>

This command deletes the selected file with import settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.import` will be loaded.

Parameters:

<Delete> string

Example: `BB:IMP:SETT:DEL 'import_1'`
deletes file `import_1`.

[:SOURce<hw>]:BB:IMPort:SETTing:LOAD <Load>

This command loads the selected file with import settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.import` will be loaded.

Setting parameters:

<Load>	string
Example:	`BB:IMP:SETT:LOAD 'import_1'` loads file `import_1`.
Usage:	Setting only

[:SOURce<hw>]:BB:IMPort:SETTing:STORe <Store>

This command stores the current import settings into the selected file. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. Only the file name has to be entered. Import settings are stored as files with the specific file extensions `*.import`.

Setting parameters:

<Store>	string
Example:	`BB:IMP:SETT:STOR 'import_1'` stores the current import settings into file `import_1`.
Usage:	Setting only

[:SOURce<hw>]:BB:IMPort:SETTing:STORe:FAST <Fast>

Determines whether the instrument performs an absolute or a differential storing of the settings.

Enable this function to accelerate the saving process by saving only the settings with values different to the default ones.

Note: This function is not affected by the "Preset" function.

Parameters:

<Fast>	0 | 1 | OFF | ON *RST: ON

[:SOURce<hw>]:BB:IMPort:SRATe:VARiation?

This command queries the sample rate variation.

Return values:

<Variation>	float
Example:	`BB:IMPort:SRATe:VARiation?`
Usage:	Query only

[:SOURce<hw>]:BB:IMPort:STATe <State>

The command enables or disables the import of IQ data.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: OFF

Example:

```
BB:IMP:STAT ON
```

enables the import of IQ data.

7.9.2 Filter/Clipping Settings

[:SOURce<hw>]:BB:IMPort:CLIPping:LEVel <Level>

The command sets the limit for level clipping. This value indicates at what point the signal is clipped. It is specified as a percentage, relative to the highest level. 100% indicates that clipping does not take place.

Level clipping is activated with the command `SOUR:BB:IMP:CLIP:STAT ON`.

Parameters:

<Level> float

Range: 0 PCT to 100 PCT
Increment: 1 PCT
*RST: 100 %
Default unit: PCT

Example:

```
BB:IMP:CLIP:LEV 80PCT
```

sets the limit for level clipping to 80% of the maximum level.

```
BB:IMP:CLIP:STAT ON
```

activates level clipping.

[:SOURce<hw>]:BB:IMPort:CLIPping:MODE <Mode>

The command sets the method for level clipping (Clipping).

Parameters:

<Mode> VECTor | SCALar

VECTor
The reference level is the amplitude | i+jq |.

SCALar
The reference level is the absolute maximum of the I and Q values.

*RST: VECTor

Example:

```
BB:IMP:CLIP:MODE SCAL
```

selects the absolute maximum of all the I and Q values as the reference level.

```
BB:IMP:CLIP:LEV 80PCT
```

sets the limit for level clipping to 80% of this maximum level.

```
BB:IMP:CLIP:STAT ON
```

activates level clipping.

[:SOURce<hw>]:BB:IMPort:CLIPping:STATe <State>

The command activates level clipping (Clipping). The value is defined with the command `[SOURce]:BB:IMPort:CLIPping:LEVel`, the mode of calculation with the command `[SOURce]:BB:IMPort:CLIPping:MODE`.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: OFF

Example:

```
BB:IMP:CLIP:STAT ON
```

activates level clipping.

[:SOURce<hw>]:BB:IMPort:FILTer:ILENgth <Ilength>

The command sets the impulse length (number of filter tabs).

Parameters:

<Ilength> float

Range:	1 to 128
Increment:	1
*RST:	10

Example:

```
BB:IMP:FILT:ILEN 10
```

sets the number of filter tabs to 10.

[:SOURce<hw>]:BB:IMPort:FILTer:ILENgth:AUTO <Auto>

The command acivates/deactivates the impulse length state. If activated, the most sensible parameter values are selected. The value depends on the coherence check.

Parameters:

<Auto> 0 | 1 | OFF | ON

*RST: ON

Example: `BB:IMP:FILT:ILEN:AUTO ON`
the most sensible parameters are selected automatically.

[:SOURce<hw>]:BB:IMPort:FILTer:OSAMpling <Osampling>

The command sets the upsampling factor.

Parameters:

<Osampling> float

Range: 1 to 32
*RST: 32

Example: `BB:IMP:FILT:OSAM 32`
sets the upsampling factor to 32.

[:SOURce<hw>]:BB:IMPort:FILTer:OSAMpling:AUTO <Auto>

The command acivates/deactivates the upsampling factor state. If activated, the most sensible parameter values are selected. The value depends on the coherence check. If deactivated, the values can be changed manually.

Parameters:

<Auto> 0 | 1 | OFF | ON

*RST: ON

Example: `BB:IMP:FILT:OSAM:AUTO ON`
the most sensible parameters are selected automatically.

[:SOURce<hw>]:BB:IMPort:FILTer:TYPE <Type>

The command selects the filter type.

Parameters:

<Type> RCOSine | COSine | GAUSs | LGAuss | CONE | COF705 | COEQualizer | COFequalizer | C2K3x | APCO25 | SPHase | RECTangle | PGAuss | LPASs | LPASSEVM | DIRac | ENPShape | EWPShape

*RST: GAUSs (if layer mode OFDM), COSine (if layer mode CCK or PBCC)

Example: `BB:IMP:FILT:TYPE COS`
selects the Cosine filter type.

[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:COSine <Cosine>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:GAUSs <Gauss>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:LPASs <Lpass>

[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:LPASSEVM <Lpassevm>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:PGAuss <Pgauss>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:RCOSine <Rcosine>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:SPHase <Sphase>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:APCO25 <Apco25>

The command sets the roll-off factor for filter type APCO25.

Parameters:

<Apco25> float

Range: 0.05 to 0.99
Increment: 0.01
*RST: 0.20

<Cosine>
Range: 0.0 to 0.99
Increment: 0.35
*RST: 0.01

<Gauss>
Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.3

<Lpass>
Range: 0.05 to 2.0
Increment: 0.01
*RST: 0.5

<LpassEvm>
Range: 0.05 to 2.0
Increment: 0.01
*RST: 0.5

<Pgauss>
Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.3

<Rcosine>
Range: 0.0 to 0.99
Increment: 0.01
*RST: 0.35

<Sphase>
Range: 0.15 to 2.5
Increment: 0.01
*RST: 2

Example:

```
BB:IMP:FILT:PAR:APCO25 0.2
```

sets the roll-off factor to 0.2 for filter type APCO25.

[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:COSine:COFS <Cofs>

The command sets the "cut of frequency shift" value for the Cosine filter type.

Parameters:

<Cofs> float

Range: -0.1 to 1.0
*RST: 0.5

Example: `BB:IMP:FILT:PAR:COS:COFS 0.04`
the "cut of frequency shift” value is set to 0.04.

7.9.3 Marker Settings

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:MODE <Mode>

The command defines the signal for the selected marker output.

Parameters:

<Mode> RESTart | PULSe | PATTern | RATio

RESTart
A brief marker signal is generated at the start of the waveform.

PULSe
A pulsed marker signal is generated. The pulse frequency (= symbol rate/divider) is defined with the `SOUR:IMP:TRIG:OUTP:PULSe:DIVider` command and can be queried with the `SOUR:BB:IMP:TRIG:OUTP:PULSe:FREQuency` command.

PATTern
A marker signal is generated with the aid of a user-definable bit pattern. The bit pattern is entered with the command `SOURce:BB:IMP:TRIGger:OUTPut:PATTern`. The bit pattern is a maximum of 32 bits long.

RATio
A regular marker signal corresponding to the Time Off / Time On specifications in the commands `SOURce:BB:IMP:TRIGger:OUTPut:OFFT` and `SOURce:BB:IMP:TRIGger:OUTPut:ONT` is generated.

*RST: RESTart

Example: `BB:IMP:TRIG:OUTP2:MODE PULS`
selects the pulsed marker signal on output MARKER 2.

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:ONTime <Ontime>
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:OFFTime <Offtime>

The command sets the number of symbols in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:IMPort:TRIGger:OUTPut:MODE RATio` on the marker outputs is OFF.

Parameters:

<Offtime> float

Range:	1 to 2^24-1
Increment:	1
*RST:	1

Example:

```
BB:IMP:TRIG:OUTP2:OFFT 20
```

sets an OFF time of 20 symbols for marker signal 2.

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PATTern <Pattern>

The command defines the bit pattern used to generate the marker signal in the setting `SOURce:BB:IMP:TRIGger:OUTPut:MODE PATTern`. 0 is marker off, 1 is marker on.

Parameters:

<Pattern> integer

Range:	#B0,1 to #B111
*RST:	#B0,1

Example:

```
BB:IMP:TRIG:OUTP2:PATT #B000000011111111,15
```

sets a bit pattern.

```
BB:IMP:TRIG:OUTP2:MODE PATT
```

activates the marker signal according to a bit pattern on output MARKER 2.

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PULSe:DIVider <Divider>

The command sets the divider for the Pulse marker mode (`SOUR:BB:IMP:TRIG:OUTP:MODE PULSe`). The resulting pulse frequency is derived by dividing the symbol rate by the divider.

Parameters:

<Divider> float

Range:	2 to 2^10
Increment:	1
*RST:	2

Example:

```
BB:IMP:TRIG:OUTP2:PULS:DIV 2
```

sets the divider to 2 for the marker signal on output MARKER 2.

```
BB:IMP:TRIG:OUTP2:FREQ?
```

queries the resulting pulse frequency of the marker signal.

Response:

```
66 000
```

the resulting pulse frequency is 66 kHz.

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PULSe:FREQuency?

The command queries the pulse frequency of the pulsed marker signal in the setting `SOURce:BB:IMP:TRIGger:OUTPut:MODE PULSe`. The pulse frequency is derived by dividing the symbol rate by the divider.

Return values:

<Frequency> float

Example:

`BB:IMP:TRIG:OUTP2:PULS:DIV 2`

sets the divider marker signal on output MARKER 2 to the value 2.

`BB:IMP:TRIG:OUTP2:MODE PULS`

enables the pulsed marker signal.

`BB:IMP:TRIG:OUTP2:PULS:FREQ?`

queries the pulse frequency of the marker signal.

Response:

`33 000`

the resulting pulse frequency is 33 kHz.

Usage: Query only

7.10 SOURce:BB:MCCW Subsystem

This subsystem contains the commands for setting the Multi Carrier CW signals.

7.10.1 Suffixes

`SOURce<hw>`

For one-path instruments, the keyword SOURce is optional and can be omitted.

The numeric suffix to `SOURce` distinguishes between signal generation for path A and path B in the case of two-path instruments:

- `SOURce[1]` = path A
  The keyword `SOURce` is optional and can be omitted
- `SOURce2` = path B
  The keyword `SOURce` is mandatory, i.e. the command must contain the keyword with suffix 2.

`OUTPut<ch>`

The numeric suffix to `OUTPut` distinguishes between the available markers.

7.10.2 General Settings and Carrier Setup Settings

[:SOURce<hw>]:BB:MCCW:STATe <State>

The command turns on the Multi Carrier CW signal. Any other digital standards or digital modulation that may be in the ON state will be automatically turned OFF.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: OFF

Example: `BB:MCCW:STAT ON`

switches on generation of the Multi Carrier CW signal.

[:SOURce<hw>]:BB:MCCW:PRESet

The command sets all multi carrier signal parameters to their default values.

Example: `BB:MCCW:PRESet`

resets the Multi Carrier settings to default values.

Usage: Event

[:SOURce<hw>]:BB:MCCW:CARRier:COUNt <Count>

The command sets the number of carriers in the Multi Carrier CW signal. The total bandwidth is calculated as (Number of carriers - 1) * Carrier spacing and must not exceed the system bandwidth of the instrument (see data sheet).

The carrier spacing (`:BB:MCCW:CARRier:SPACing`) is reduced if the total bandwidth is not respected when entering the number of carriers.

The number of carriers entered therefore defines the maximum carrier spacing (`:BB:MCCW:CARRier:SPACing`).

Parameters:

<Count> integer

Range: 1 to 8192

*RST: 64

Example:
`BB:MCCW:CARR:COUN 10`
sets 10 CW carriers for the multi carrier signal.

[:SOURce<hw>]:BB:MCCW:CARRier:LIST:PHASe <Phas0[,Phas1..]>
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:PHASe? [, <Start>[, <Count>]]

The command sets the start phase of the carrier with the aid of a value list. The first value in the list is assigned to the carrier with index 0, the second value to the carrier with index 1, etc. The maximum length corresponds to the maximum number of multi carriers. There is no need to enter all the values every time. Values not set by the value list are set with the default values provided they have already been explicitly set by a previous command. If this is the case, the values continue to apply until overwritten.

Setting parameters:

<Phas0[,Phas1..]> float

Range:	0 to 360
Increment:	0.01
*RST:	0
Default unit:	DEG

Query parameters:

<Start> integer

Range: 0 to lastCarrier

<Count> integer

Range: 1 to lastCarrier

Return values:

<Phas[,Phas..]> float

Example:
`BB:MCCW:CARR:LIST:PHAS 1,20,30,40,50,60`
sets a start phase for carriers 0, 1, 2, 3 and 4.
`BB:MCCW:CARR:LIST:PHAS? 2,3`
queries the phase of carrier 1, 2 and 3.
Response: `20,30,40`

[:SOURce<hw>]:BB:MCCW:CARRier:LIST:POWer <Pow0[,Pow1..]>
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:POWer? <Start>, <Count>

The command sets the power of the carrier with the aid of a value list. The first value in the list is assigned to the carrier with index 0, the second value to the carrier with index 1, etc. The maximum length corresponds to the maximum number of multi carriers. There is no need to enter all the values every time. Values not set by the value list are set with the default values provided they have already been explicitly set by a previous command. If this is the case, the values continue to apply until overwritten.

Setting parameters:

<Pow0[,Pow1..]> float

Increment:	0.01
*RST:	0 dB

Query parameters:

<Start>	integer	
	Range:	0 to lastCarrier
<Count>	integer	
	Range:	1 to lastCarrier

Return values:

<Pow[,Pow..]>	float

Example:

```
BB:MCCW:CARR:LIST:POW -65 dB, -30 dB, -50
dB,...
```

sets the power of carrier 0 to -65 dB, carrier 1 to -30 dB and so on.

```
BB:MCCW:CARR:LIST:POW? 2,3
```

queries the power of carrier 1 and 2.

Response: `-30,-50`

[:SOURce<hw>]:BB:MCCW:CARRier:LIST:STATe <Stat0[,Stat1..]>]
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:STATe? <Start>, <Count>

The command switches the carrier on or off with the aid of a value list. The first value in the list is assigned to the carrier with index 0, the second value to the carrier with index 1, etc. The maximum length corresponds to the maximum number of multi carriers. There is no need to enter all the values every time. Values not set by the value list are set with the default values provided they have already been explicitly set by a previous command. If this is the case, the values continue to apply until overwritten.

Setting parameters:

<Stat0[,Stat1..]>	ON | OFF[,ON | OFF,...]	
	*RST:	ON

Query parameters:

<Start>	integer	
	Range:	0 to lastCarrier
<Count>	integer	
	Range:	1 to lastCarrier

Return values:

<State[,State..]>	select

Example:

```
BB:MCCW:CARR:LIST:STAT 1,0,1,1,0,0....
```

switches carrier 0 on, carrier 1 off, etc.

```
BB:MCCW:CARR:LIST:STAT? 2,4
```

queries the states of carrier 2, 3, 4 and 5.

Response: `0,1,1,0`

[:SOURce<hw>]:BB:MCCW:CARRier:PHASe <Phase>

The command sets the start phase of the selected carrier.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:MCCW:CFACtor:MODE OFF`).

Parameters:

<Phase> float

Sets the start phase of the selected carrier.

Range: 0 to 359.99
Increment: 0.01
*RST: 0
Default unit: DEG

Example:

```
BB:MCCW:CARR:PHAS 15,90
```

sets a start phase of 90 DEG for carrier 15.

[:SOURce<hw>]:BB:MCCW:CARRier:POWer <Power>

The command sets the power of the selected carrier.

Parameters:

<Power> float

Sets the power of the selected carrier.

Range: -80 to 0
Increment: 0.01
*RST: 0

Example:

```
BB:MCCW:CARR:POW 15, -50 dB
```

sets the power of carrier 15 to -50 dB.

[:SOURce<hw>]:BB:MCCW:CARRier:SPACing <Spacing>

The command sets the carrier spacing.

The carriers are generated symmetrically around the center carrier. The total bandwidth is calculated as ("Number of carriers" - 1) * "Carrier spacing "and must not exceed the system bandwidth of the instrument (see data sheet).

The maximum carrier spacing that can be set is dependent on the chosen number of carriers

The maximum carrier spacing is automatically reduced so that the maximum total bandwidth is not exceeded on entering the number of carriers (`:BB:MCCW:CARRier:COUNt`).

Parameters:

<Spacing> float

Range: 0 to 120E6
Increment: 0.01
*RST: 10E3

Example:

```
BB:MCCW:CARR:SPAC 10 MHz
```

sets a carrier spacing of 10 MHz.

[:SOURce<hw>]:BB:MCCW:CARRier:STATe <State>

The command switches the selected carrier on or off.

The counting in remote control differs from the numbers in the carrier table. Index 0 corresponds to number 1 (first line) in the table. Therefore, switching the state of the channel via remote control always switches the state of <channel index> + 1 in the table.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `BB:MCCW:CARR:STAT 15, ON`

switches carrier 16 on.

[:SOURce<hw>]:BB:MCCW:CFACtor <CFactor>

The command sets the desired crest factor for the multi carrier signal on selection of the command `SOUR:BB:MCCW:CFAC:MODE SLOW`.

Parameters:

<CFactor> float

Range: 0 to 30
Increment: 0.01
*RST: 3

Example: `BB:MCCW:CFAC:MODE SLOW`

sets the Target Crest optimization mode.

`BB:MCCW:CFAC 10 dB`

sets the desired crest factor to 10 dB.

[:SOURce<hw>]:BB:MCCW:CFACtor:ACTual?

Queries the actual Crest Factor for `BB:MCCW:CFAC:MODE SLOW`.

Return values:

<Actual> float

Range: 0 to 100
Increment: 0.01
*RST: 3

Example: `SOUR:BB:MCCW:CFAC:MODE SLOW`

`BB:MCCW:CFAC:ACT?`

Response: `3`

Usage: Query only

[:SOURce<hw>]:BB:MCCW:CFACtor:MODE <Mode>

The command sets the mode by which automatic settings will minimize the crest factor or hold it at a chosen value.

Parameters:

<Mode> OFF | CHIRp | SLOW

OFF
Crest factor optimization is switched off. The carrier PHASE setting has an effect.

CHIRp
The crest factor is very rapidly optimized to < 3 dB for multi carrier signals so that all carriers are switched on and have the same amplitude. The computation time is independent of the number of carriers. In other carrier configurations the achievable crest factor is worse.

SLOW
The crest factor entered using `SOURceBB:MCCW:CFACtor` is maintained for all carrier configurations by means of automatic settings. The computation time depends on the number of carriers and on the crest factor that has been set. Computation time increases only when the number of carriers exceeds 256 and the crest factor is above 4 dB.

*RST: CHIRp

Example: `BB:MCCW:CFAC:MODE OFF`
switches off automatic crest factor optimization.
The setting `SOURBB:MCCW:CARR:PHAS` has an effect.

[:SOURce<hw>]:BB:MCCW:CLOCk?

The command queries the output clock rate. The output clock rate depends on the number of carriers an the selected carrier spacing.

Return values:

<Clock> float

Range: 0 to Max
Increment: 1E-3
*RST: 0

Example: `BB:MCCW:CLOC?`
queries the output clock rate.
Response: `256 000 000`
the output clock rate is 256 MHz.

Usage: Query only

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:EXECute

The command adopts the settings for the carrier range which has been defined using the `:BB:MCCW:EDIT:CARR:...` commands.

Example:

```
BB:MCCW:EDIT:CARR:STAR 4
```

the carrier range starts at carrier 4.

```
BB:MCCW:EDIT:CARR:STOP 400
```

the carrier range stops at carrier 400.

```
BB:MCCW:EDIT:CARR:STAT ON
```

sets all the carriers in the carrier range to ON.

```
BB:MCCW:EDIT:CARR:EXEC
```

adopts the settings for all the carriers in the carrier range.

```
BB:MCCW:STAT
```

starts generation of the multi carrier signal. Carriers 4 to 400 are in the ON state.

Usage: Event

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:PHASe:STEP <Step>

The command sets the step width by which the start phases of the carriers in the defined carrier range will be incremented.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:MCCW:CFACtor:MODE OFF`).

Parameters:

<Step> float

Range:	-359.99 to 359.99
Increment:	0.01
*RST:	0

Example:

```
BB:MCCW:EDIT:CARR:PHAS 90 DEG
```

sets a start phase of 90° for the carriers in the carrier range.

```
BB:MCCW:EDIT:CARR:PHAS:STEP 1 DEG
```

the start phase is incremented by 1° for each carrier, i.e. the first carrier has a start phase of 90°, the second a start phase of 91°, etc.

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:PHASe[:STARt] <Start>

The command sets the start phase for the individual carriers in the defined carrier range. If the command `:BB:MCCW:EDIT:CARR:PHAS:STEP` is used to define a step width, the phase entered here applies only to the starting carrier. The phases of the remaining carriers are stepped up or down by the phase value specified in the `:BB:MCCW:EDIT:CARR:PHAS:STEP` command.

The phase settings are only valid if optimization of the crest factor is disabled (`SOURce:BB:MCCW:CFACtor:MODE OFF`).

Parameters:

<Start> float

Range:	0 to 359.99
Increment:	0.01
*RST:	0

Example: `BB:MCCW:EDIT:CARR:PHAS 90 DEG`
sets a start phase of 90° for the carriers in the carrier range.

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:POWer:STEP <Step>

The command sets the step width by which the starting power of the carriers in the defined carrier range will be incremented.

Parameters:

<Step> float

Range:	-80 to 80
Increment:	0.01
*RST:	0

Example: `BB:MCCW:EDIT:CARR:POW -80dB`
sets a power of -80 dB for the carriers in the carrier range.
`BB:MCCW:EDIT:CARR:POW:STEP 1 dB`
the power phase is incremented by dB for each carrier, i.e. the first carrier has -80 dB, the second -79 dB, etc.

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:POWer[:STARt] <Start>

The command sets the power for the individual carriers in the defined carrier range. If the command `:BB:MCCW:EDIT:CARR:POW:STEP` is used to define a step width, the power entered here applies only to the starting carrier. The power of the remaining carriers is stepped up or down by the power specified in the `:BB:MCCW:EDIT:CARR:POW:STEP` command.

Parameters:

<Start> float

Range:	-80 to 0
Increment:	0.01
*RST:	0

Example: `BB:MCCW:EDIT:CARR:POW -50 dB`
sets the power of the carrier to -50 dB.

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STARt <Start>

The command selects the first carrier in the carrier range to which the settings with the `:BB:MCCW:EDIT:CARR:...` commands shall apply.

Parameters:

<Start> integer

Range:	0 to 8191
Increment:	1
*RST:	0

Example: `BB:MCCW:EDIT:CARR:STAR 4`
the carrier range starts at carrier 4.

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STATe <State>

The command switches all the carriers in the selected carrier range on or off.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 1

Example:

```
BB:MCCW:EDIT:CARR:STAT ON
```

sets all the carriers in the carrier range to ON.

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STOP <Stop>

The command selects the last carrier in the carrier range to which the settings with the `:BB:MCCW:EDIT:CARR:...` commands shall apply.

Parameters:

<Stop> integer

Range: 0 to 8191
Increment: 1
*RST: 0

Example:

```
BB:MCCW:EDIT:CARR:STOP 40
```

the carrier range stops at carrier 40.

7.10.3 Marker Settings

This section lists the remote control commands, necessary to configure the markers.

The marker delay settings are available for R&S SMx and R&S AMU instruments only.

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:MODE <Mode>

The command defines the signal for the selected marker output.

Parameters:

<Mode> RESTart | PULSe | PATTern | RATio

RESTart
A brief marker signal is generated at the start of the waveform.

PULSe
A pulsed marker signal is generated. The pulse frequency (= sample rate/divider) is defined with the `SOUR:BB:MCCW:TRIG:OUTP:PULSe:DIVider` command and can be queried with the `SOUR:BB:MCCW:TRIG:OUTP:PULSe:FREQuency?` command.

PATTern
A marker signal is generated with the aid of a user-definable bit pattern. The bit pattern is entered with the aid of command `SOURce:BB:MCCW:TRIGger:OUTPut:PATTern`. The bit pattern is a maximum of 32 bits long.

RATio
A regular marker signal corresponding to the Time Off / Time On specifications in the commands `SOURce:BB:MCCW:TRIGger:OUTPut:OFFT` and `SOURce:BB:MCCW:TRIGger:OUTPut:ONT` is generated.

*RST: RESTart

Example: `BB:MCCW:TRIG:OUTP2:MODE PULS`
selects the pulsed marker signal on output MARKER 2.

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:OFFTime <OffTime>
[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:ONTime <OnTime>

The command sets the number of samples in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:MCCW:TRIGger:OUTPut:MODE RATio` on the marker outputs is ON.

Parameters:

<OnTime> integer

Range: 1 to 16777215
Increment: 1
*RST: 1

Example: `BB:MCCW:TRIG:OUTP2:ONT 20`
sets an ON time of 20 samples for marker 2.

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PATTern <Pattern>

The command defines the bit pattern used to generate the marker signal in the setting `SOURce:BB:MCCW:TRIGger:OUTPut:MODE PATTern` 0 is marker off, 1 is marker on.

Parameters:

<Pattern> <32 bit pattern>

*RST: 0

Example:

`BB:MCCW:TRIG:OUTP2:PATT #HE0F52,20`
sets a bit pattern.
`BB:MCCW:TRIG:OUTP2:MODE PATT`
activates the marker signal according to a bit pattern on output MARKER 2.

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PULSe:DIVider <Divider>

The command sets the divider for Pulse marker mode (`SOUR:BB:MCCW:TRIG:OUTP:MODE PULSe`). The resulting pulse frequency is derived by dividing the symbol rate by the divider.

Parameters:

<Divider> integer

Range: 2 to 1024
Increment: 1
*RST: 2

Example:

`BB:MCCW:TRIG:OUTP2:PULS:DIV 2`
sets the divider for the marker signal on output MARKER 2 to the value 2.
`BB:MCCW:TRIG:OUTP2:FREQ?`
queries the resulting pulse frequency of the marker signal
`Response: 66 000`
the resulting pulse frequency is 66 kHz.

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PULSe:FREQuency?

The command queries the pulse frequency of the pulsed marker signal in the setting `SOUR:BB:MCCW:TRIG:OUTP:MODE PULS`. The pulse frequency is derived by dividing the symbol rate by the divider. The divider is defined with command `SOUR:BB:MCCW:TRIG:OUTP:PULS:DIV`.

Return values:

<Frequency> float

Increment: 1E-3

Example:

`BB:MCCW:TRIG:OUTP2:PULS:DIV 4`
sets the divider for the marker signal on output MARKER 2 to the value 4.
`BB:MCCW:TRIG:OUTP2:MODE PULS`
enables the pulsed marker signal
`BB:MCCW:TRIG:OUTP2:PULS:FREQ?`
queries the pulse frequency for the marker signal.
`Response: "33 000"`
the resulting pulse frequency is 33 kHz.

Usage: Query only

8 Error Messages

This chapter describes the error messages of the R&S WinIQSIM2. The error messages are output in the "Info" line on the screen and entered in the error/event queue of the status reporting system.

A great variety of different messages such as status messages, error messages, warnings or information are displayed in the header field of the screen. Some error messages require that the error must be eliminated before correct instrument operation can be ensured. The info window with a list of current messages and a detailed description of each message can be opened with the INFO key.

In the remote control mode, error messages are entered in the error/event queue of the status reporting system and can be queried with the command `SYSTem:ERRor?`. If the error queue is empty, 0 ("No error") is returned.

8.1 Status Information

The status messages are displayed in the header section of the screen. The status information gives the user an overview of the main operating states and settings of the instrument. The states are indicated for information only and do not necessitate any action by the user. Status information is displayed between the frequency and level fields, at the left of the info line or in the info line itself.

8.1.1 Status information displayed between PEP and level fields

This chapter gives an overview of the status messages displayed between the PEP and level fields.

EXT REF
An external reference is used.

The external signal with selectable frequency and defined level must be input at the REF IN connector. It is output at the REF OUT connector.

BUSY
A setting or calculation is executed.

8.1.2 Status information displayed to the left of the Info line

This chapter gives an overview of the status messages displayed to the left of the Info line.

REMOTE
The instrument is remote controlled.

The LOCAL key switches the instrument from remote control to manual operation. The current command must be fully processed before the mode is switched, otherwise the instrument switches immediately back to remote control.

REM-LLO
The instrument is remote (**REM**ote) controlled. The LOCAL key is disabled by remote control with the command `LLO` (**L**ocal**L**ock**O**ut).

The instrument can be switched from remote control to manual operation by means of remote control only (e.g. with the Visual Basic command `CALL IBLOC (generator%)` or by `&NREN`).

LOC-LLO
For operating directly the instrument is placed from remote control to manual operation (Local State). The LOCAL key was disabled by remote control with the command `LLO` (**L**ocal**L**ock**O**ut).

With the next activating of the remote control mode, the instrument cannot be switched to manual operation by the operator. The status information changes to "REM-LLO".

The instrument can be switched to manual operation by means of remote control only (e.g. with the Visual Basic command `CALL IBLOC (generator%)`).

8.2 Error Messages

Messages indicate errors in the instrument. They are displayed in the info line in different colors depending on their importance and display duration. Errors (e.g. no calibration data) are displayed in red, information (e.g. file not found) and warnings in black. Warnings indicate less significant errors (e.g. the instrument operates outside specified data).

8.2.1 Volatile messages

Volatile messages report automatic settings in the instrument (e.g. switching off of incompatible types of modulation) or on illegal entries that are not accepted by the instrument (e.g. range violations). They are displayed in the info line on a yellow background. They are displayed on top of status information or permanent messages.

Volatile messages do not normally demand user actions and disappear automatically after a brief period of time. They are stored in the history, however.

8.2.2 Permanent messages

Permanent messages are displayed if an error occurs that impairs further instrument operation, e.g. a hardware fault. The error signaled by a permanent message must be eliminated before correct instrument operation can be ensured.

The message is displayed until the error is eliminated. It covers the status display in the info line. After error elimination, the message automatically disappears and is also recorded in the history.

8.3 SCPI-Error Messages

The SCPI error messages are the same in all SCPI instruments. Detailed information and an overview of all error messages as defined in SCPI standard can be found in the corresponding documentation.

The errors are assigned negative numbers. The error text being entered into the error/ event queue or being displayed is printed in bold face on the left together with the error code. Below the error text, there is an explanation as to the respective error.

8.4 Device-Specific Error Messages

The following table contains all error messages specific for the instrument in alphabetical order, as well as an explanation of the error situation. The positive error codes mark the errors specific of the instrument.

The device-specific error messages set bit 3 in the ESR register.

The index provides a list of the error messages sorted according to their error codes.

Error Code	Error	Description	Remedy
140	This modulation forces other modulations off	A modulation has been switched on which cannot be used at the same time as an already active modulation. The previous modulation has been switched off. Example: Enabling modulation GSM/ EDGE switches any active digital modulation off.	
180	Adjustment failed	Adjustment could not be executed	The adjustment data have to be generated first by an internal or external adjustment or to be loaded into the device.
182	Adjustment data missing	Adjustment data are missing.	The adjustment data have to be generated first by an internal or external adjustment or to be loaded into the instrument.
183	Adjustment data invalid	Adjustment data are invalid and must be restored.	The adjustment data have to be generated again by an internal or external adjustment or to be loaded into the instrument.

Error Code	Error	Description	Remedy
200	Cannot access hardware	The data transmission to a module was unsuccessful.	The module is not installed, not properly installed or missing.
201	Hardware revision out of date	A later version of certain parts of the instrument is necessary to execute the function selected.	The driver does not support the installed version of a module.
202	Cannot access the EEPROM	A error occurs when writing or reading a EEPROM.	The EEPROM might be defect and has to be replaced.
204	Driver initialization failed	Initialization of a driver fails when booting the instrument firmware.	The driver is not compatible with the hardware or software configuration of the instrument.
241	No current list	There is no list selected. To execute the required operation, a list has to be selected in the related menu. If no list is available, a new list must be created.	
242	Unknown list type specified	The list type selected is not valid for the required operation. For instance, the file extension for waveform list files is *.wv. It is not possible to enter another file extension when selecting a list.	Check the selected list type.
261	Waveform Protected	The selected waveform file cannot be transferred to a controller. The waveform is produced with simulation software R&S WinIQSIM2 and is protected.	
460	Cannot open file	The selected file can not be opened.	Check the path and file name.
461	Cannot write file	The file can not be written.	Check if the file is read-only.
462	Cannot read file	The file can not be read.	Check if the file contents are compatible with the file type.
463	Filename missing	The required operation cannot be executed because the file name is not specified.	A file name has to be entered when creating a new list.
464	Invalid filename extension	The file extension is not valid for the required operation.	Check the file extension. For instance, the file extension for waveform list files is `*.wv`. It is not possible to enter another file extension when storing a list.
465	File contains invalid data	The selected file contains data that is not valid for the file type. The file extension determines the data that is valid for this file type. If the file extension is changed the lists are no longer recognized and the data are therefore invalid. Example: the extension of a waveform file (= `*.wv`) was changed to `*.txt`	Check the file extension.

List of Commands

Index