soa 接口定义

用rti编译为 libSoaDataTypes.a

IDL Type	Example Entry in IDL File	RTI Code Generate Example
char	char member;	`char member() const OMG_NOEXCEPT;` `void member(char value);`
wchar	wchar member;	`DDS_Wchar member() const OMG_NOEXCEPT;` `void member(DDS_Wchar value);`
octet	octet member;	`uint8_t member() const OMG_NOEXCEPT;` `void member(uint8_t value);`
short	short member;	`int16_t member() const OMG_NOEXCEPT;` `void member(int16_t value);`
unsigned short	unsigned short member;	`int16_t member() const OMG_NOEXCEPT;` `void member(int16_t value);`
long	long member;	`int32_t member() const OMG_NOEXCEPT;` `void member(int32_t value);`
unsigned long	unsigned long member;	`uint32_t member() const OMG_NOEXCEPT;` `void member(uint32_t value);`
long long	long long member;	`rti::core::int64 member() const OMG_NOEXCEPT;` `void member(rti::core::int64 value);`
unsigned long long	unsigned long long member;	`rti::core::uint64 member);` `rti::core::uint64 member() const OMG_NOEXCEPT;`
float	float member;	`float member() const OMG_NOEXCEPT;` `void member(float value);`
double	double member;	`double member() const OMG_NOEXCEPT;` `void double_member(double value);`
long double	long double member;	`rti::core::LongDouble& member() OMG_NOEXCEPT;` `const rti::core::LongDouble& member() const OMG_NOEXCEPT;` `void member(const rti::core::LongDouble& value);`
pointer	long * member;	`int32_t * member() const OMG_NOEXCEPT;` `void member(int32_t * value);`
boolean	boolean member;	`bool boolean_member() const OMG_NOEXCEPT;` `void boolean_member(bool value);`

IDL Type	Example Entry in IDL File	RTI Code Generate Example
enum	enum PrimitiveEnum { ENUM1, ENUM2, ENUM3 }; enum PrimitiveEnum { ENUM1 = 10, ENUM2 = 20, ENUM3 = 30 };	struct PrimitiveEnum_def { enum type { ENUM1, ENUM2, ENUM3 }; }; typedef dds::core::safe_enum<PrimitiveEnum_def> PrimitiveEnum; struct PrimitiveEnum_def { enum type { ENUM1 = 10, ENUM2 = 20, ENUM3 = 30 }; }; typedef dds::core::safe_enum<PrimitiveEnum_def> PrimitiveEnum;
constant	const short SIZE = 5;	static const int16_t SIZE = 5;
struct	char char_member;	char char_member() const OMG_NOEXCEPT; void char_member(char value);
union	union PrimitiveUnion switch (long){ case 1: short short_member; default: long long_member; };	class PrimitiveUnion { public: int32_t *d() const ;* *void d(int32_t value);* *int16_t short_member() const ;* *void short_member(int16_t value);* *int32_t long_member() const ;* *void long_member(int32_t value);* *static int32_t default_discriminator();* *private:* *int32_t m_d;* *struct Union* { int16_t m_short_member_; int32_t m_long_member_; Union_(); Union_(int16_t short_member, int32_t long_member); }; Union_ m_u_; };
typedef	typedef short TypedefShort;	typedef int16_t TypedefShort; struct TypedefShort_AliasTag_t {};

IDL Type	Example Entry in IDL File	RTI Code Generate Example
array of above types	struct OneDArrayStruct { short short_array[2]; };	struct TwoDArrayStruct { short short_array[1][2]; }; class OneDArrayStruct { public: dds::core::array<int16_t, 2>& short_array() OMG_NOEXCEPT; const dds::core::array<int16_t, 2>& short_array() const OMG_NOEXCEPT; void short_array(const dds::core::array<int16_t, 2>& value); }; class TwoDArrayStruct { public: dds::core::array<dds::core::array<int16_t, 2>, 1>& short_array() OMG_NOEXCEPT; const dds::core::array<dds::core::array<int16_t, 2>, 1>& short_array() const OMG_NOEXCEPT; void short_array(const dds::core::array<dds::core::array<int16_t, 2>, 1>& value); };
bounded sequence of above types	struct SequenceStruct { sequence<short,4> short_sequence; };	class SequenceStruct { public: dds::core::vector<int16_t>& short_sequence() OMG_NOEXCEPT; const dds::core::vector<int16_t>& short_sequence() const OMG_NOEXCEPT; void short_sequence(const dds::core::vector<int16_t>& value); };
unbounded sequence of above types	struct SequenceStruct { sequence short_sequence; };	class SequenceStruct { public: dds::core::vector<int16_t>& short_sequence() OMG_NOEXCEPT; const dds::core::vector<int16_t>& short_sequence() const OMG_NOEXCEPT; void short_sequence(const dds::core::vector<int16_t>& value); };

IDL Type	Example Entry in IDL File	RTI Code Generate Example
array of sequences	`struct ArraysOfSequences{` `  sequence<short,4>` `sequences_array[2];` `};`	`class ArraysOfSequences {` `public:` `dds::core::array<dds::core::vector<int16_t>,` `2>& sequences_array() OMG_NOEXCEPT;` `  const` `dds::core::array<dds::core::vector<int16_t>,` `2>& sequences_array() const OMG_NOEXCEPT;` `  void sequences_array(const` `dds::core::array<dds::core::vector<int16_t>,` `2>& value);` `};`
sequence of arrays	`typedef short` `ShortArray[2];` `struct SequenceofArrays {` `  sequence<ShortArray,2>` `  arrays_sequence;` `};`	`typedef dds::core::array<int16_t, 2>` `ShortArray;` `class SequenceofArrays {` `public:` `  dds::core::vector& arrays_sequence()` `OMG_NOEXCEPT;` `  const dds::core::vector&` `arrays_sequence() const OMG_NOEXCEPT;` `  void arrays_sequence(const` `dds::core::vector& value);` `};`
sequence of sequences	`typedef sequence<short,4>` `  ShortSequence;` `struct` `SequencesOfSequences{` `sequence<ShortSequence,2>` `sequences_sequence;` `};`	`typedef dds::core::vector<int16_t>` `ShortSequence;` `class SequencesOfSequences {` `public:` `  dds::core::vector& sequences_sequence()` `OMG_NOEXCEPT;` `  const dds::core::vector&` `sequences_sequence() const OMG_NOEXCEPT;` `  void sequences_sequence(const` `dds::core::vector& value);` `};`

IDL Type	Example Entry in IDL File	RTI Code Generate Example
bounded string	`struct PrimitiveStruct { string<20> string_member; };`	`class PrimitiveStruct { public: dds::core::string& string_member() OMG_NOEXCEPT; const dds::core::string& string_member() const OMG_NOEXCEPT; void string_member(const dds::core::string& value); };`
unbounded string	`struct PrimitiveStruct { string string_member; };`	`class PrimitiveStruct { public: dds::core::string& string_member() OMG_NOEXCEPT; const dds::core::string& string_member() const OMG_NOEXCEPT; void string_member(const dds::core::string& value); };`
bounded wstring	`struct PrimitiveStruct { wstring<20> wstring_member; };`	`class PrimitiveStruct { public: dds::core::wstring& wstring_member() OMG_NOEXCEPT; const dds::core::wstring& wstring_member() const OMG_NOEXCEPT; void wstring_member(const dds::core::wstring& value); };`
unbounded wstring	`struct PrimitiveStruct { wstring wstring_member; };`	`class PrimitiveStruct { public: dds::core::wstring& wstring_member() OMG_NOEXCEPT; const dds::core::wstring& wstring_member() const OMG_NOEXCEPT; void wstring_member(const dds::core::wstring& value); };`

IDL Type	Example Entry in IDL File	RTI Code Generate Example
module	`module PackageName { struct Foo { long field; }; };`	`namespace PackageName { class Foo { public: int32_t field() const OMG_NOEXCEPT; void field(int32_t value); }; };`
valuetype	`valuetype MyBaseValueType { public long member; }; valuetype MyValueType: MyBaseValueType { public short * member2; };`	`class MyBaseValueType { public: int32_t member() const OMG_NOEXCEPT; void member(int32_t value); }; class MyValueType : public MyBaseValueType { public: int16_t * member2() const OMG_NOEXCEPT; void member2(int16_t * value); };`