

C++ Software Transactional Memory

v 0.0.1

Generated by Doxygen 1.8.11

Contents

1	OSTM C++ Software Transactional Memory	1
1.1	Object Based Software Transactional Memory.	1
1.1.1	Brief. Download the zip file from the provided link in the web-site, that contains the libostm.so, TM.h, TX.h, OSTM.h files.	1
1.1.2	Step 1: Download the archive file.	1
1.1.3	Step 2: Unzip in the target destination.	1
1.1.4	Step 3: Copy the shared library (libostm.so) to the operating system folder where the other shared library are stored.	1
1.1.5	Step 4: Achieve the required class hierarchy between the OSTM library and your own class structure.	1
1.1.6	Step 5: Create an executable file as you linking together the TM.h, TX.h, OSTM.h files with your own files.	2
1.1.7	Step 6: Now your application use transactional environment, that guarantees the consistency between object transactions.	2
1.1.8	Step 7: Run the application.	2
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7

5	Class Documentation	9
5.1	AIB Class Reference	9
5.1.1	Detailed Description	10
5.1.2	Constructor & Destructor Documentation	10
5.1.2.1	AIB()	10
5.1.2.2	AIB(int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)	10
5.1.2.3	AIB(std::shared_ptr< BANK > obj, int _version, int _unique_id)	10
5.1.2.4	AIB(const AIB &orig)	10
5.1.2.5	~AIB()	10
5.1.3	Member Function Documentation	10
5.1.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	10
5.1.3.2	GetAccountNumber() const	11
5.1.3.3	GetAddress() const	11
5.1.3.4	GetBalance() const	11
5.1.3.5	getBaseCopy(std::shared_ptr< OSTM > object)	11
5.1.3.6	GetFirstName() const	12
5.1.3.7	GetFullname() const	12
5.1.3.8	GetLastName() const	12
5.1.3.9	operator=(const AIB &orig)	12
5.1.3.10	SetAccountNumber(int accountNumber)	12
5.1.3.11	SetAddress(std::string address)	13
5.1.3.12	SetBalance(double balance)	13
5.1.3.13	SetFirstName(std::string firstName)	13
5.1.3.14	SetFullname(std::string fullname)	13
5.1.3.15	SetLastName(std::string lastName)	13
5.1.3.16	toString()	13
5.2	BANK Class Reference	14
5.2.1	Detailed Description	14
5.2.2	Constructor & Destructor Documentation	14
5.2.2.1	BANK()	14

5.2.2.2	BANK(int _version, int _unique_id)	14
5.2.2.3	BANK(const BANK &orig)	15
5.2.2.4	~BANK()	15
5.2.3	Member Function Documentation	15
5.2.3.1	GetAccountNumber() const	15
5.2.3.2	GetAddress() const	15
5.2.3.3	GetBalance() const	15
5.2.3.4	GetFirstName() const	15
5.2.3.5	GetFullname() const	15
5.2.3.6	GetLastName() const	16
5.2.3.7	SetAccountNumber(int accountNumber)	16
5.2.3.8	SetAddress(std::string address)	16
5.2.3.9	SetBalance(double balance)	16
5.2.3.10	SetFirstName(std::string firstName)	16
5.2.3.11	SetFullname(std::string fullname)	16
5.2.3.12	SetLastName(std::string lastName)	16
5.3	BOA Class Reference	17
5.3.1	Detailed Description	17
5.3.2	Constructor & Destructor Documentation	17
5.3.2.1	BOA()	17
5.3.2.2	BOA(int accountNumber, double balance, std::string firstName, std::string last↵ Name, std::string address)	18
5.3.2.3	BOA(std::shared_ptr< BANK > obj, int _version, int _unique_id)	18
5.3.2.4	BOA(const BOA &orig)	18
5.3.2.5	~BOA()	18
5.3.3	Member Function Documentation	18
5.3.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	18
5.3.3.2	GetAccountNumber() const	19
5.3.3.3	GetAddress() const	19
5.3.3.4	GetBalance() const	19
5.3.3.5	getBaseCopy(std::shared_ptr< OSTM > object)	19

5.3.3.6	GetFirstName() const	19
5.3.3.7	GetFullname() const	20
5.3.3.8	GetLastName() const	20
5.3.3.9	operator=(const BOA &orig)	20
5.3.3.10	SetAccountNumber(int accountNumber)	20
5.3.3.11	SetAddress(std::string address)	20
5.3.3.12	SetBalance(double balance)	20
5.3.3.13	SetFirstName(std::string firstName)	21
5.3.3.14	SetFullname(std::string fullname)	21
5.3.3.15	SetLastName(std::string lastName)	21
5.3.3.16	toString()	21
5.4	BOI Class Reference	21
5.4.1	Detailed Description	22
5.4.2	Constructor & Destructor Documentation	22
5.4.2.1	BOI()	22
5.4.2.2	BOI(int accountNumber, double balance, std::string firstName, std::string last↵ Name, std::string address)	22
5.4.2.3	BOI(std::shared_ptr< BOI > obj, int _version, int _unique_id)	23
5.4.2.4	BOI(const BOI &orig)	23
5.4.2.5	~BOI()	23
5.4.3	Member Function Documentation	23
5.4.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	23
5.4.3.2	GetAccountNumber() const	23
5.4.3.3	GetAddress() const	24
5.4.3.4	GetBalance() const	24
5.4.3.5	getBaseCopy(std::shared_ptr< OSTM > object)	24
5.4.3.6	GetFirstName() const	24
5.4.3.7	GetFullname() const	24
5.4.3.8	GetLastName() const	25
5.4.3.9	operator=(const BOI &orig)	25
5.4.3.10	SetAccountNumber(int accountNumber)	25

5.4.3.11	SetAddress(std::string address)	25
5.4.3.12	SetBalance(double balance)	25
5.4.3.13	SetFirstName(std::string firstName)	25
5.4.3.14	SetFullname(std::string fullname)	26
5.4.3.15	SetLastName(std::string lastName)	26
5.4.3.16	toString()	26
5.5	CARLOW_W Class Reference	26
5.5.1	Detailed Description	27
5.5.2	Constructor & Destructor Documentation	27
5.5.2.1	CARLOW_W()	27
5.5.2.2	CARLOW_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	28
5.5.2.3	CARLOW_W(std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id)	28
5.5.2.4	CARLOW_W(const CARLOW_W &orig)	28
5.5.2.5	~CARLOW_W()	28
5.5.3	Member Function Documentation	28
5.5.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	28
5.5.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	29
5.5.3.3	GetNumber_of_alcatel()	29
5.5.3.4	GetNumber_of_huawei()	29
5.5.3.5	GetNumber_of_iphones()	29
5.5.3.6	GetNumber_of_nokia()	29
5.5.3.7	GetNumber_of_samsung()	30
5.5.3.8	GetNumber_of_sony()	30
5.5.3.9	GetShop_address()	30
5.5.3.10	GetShop_name()	30
5.5.3.11	operator=(const CARLOW_W &orig)	30
5.5.3.12	SetNumber_of_alcatel(int _number_of_alcatel)	30
5.5.3.13	SetNumber_of_huawei(int _number_of_huawei)	31
5.5.3.14	SetNumber_of_iphones(int _number_of_iphones)	31
5.5.3.15	SetNumber_of_nokia(int _number_of_nokia)	31

5.5.3.16	SetNumber_of_samsung(int _number_of_samsung)	31
5.5.3.17	SetNumber_of_sony(int _number_of_sony)	31
5.5.3.18	SetShop_address(std::string _shop_address)	31
5.5.3.19	SetShop_name(std::string _shop_name)	32
5.5.3.20	toString()	32
5.6	CARPHONE_WAREHOUSE Class Reference	32
5.6.1	Detailed Description	33
5.6.2	Constructor & Destructor Documentation	33
5.6.2.1	CARPHONE_WAREHOUSE()	33
5.6.2.2	CARPHONE_WAREHOUSE(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	33
5.6.2.3	CARPHONE_WAREHOUSE(std::shared_ptr< WAREHOUSE > obj, int _↵ version, int _unique_id)	33
5.6.2.4	CARPHONE_WAREHOUSE(const CARPHONE_WAREHOUSE &orig)	34
5.6.2.5	~CARPHONE_WAREHOUSE()	34
5.6.3	Member Function Documentation	34
5.6.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	34
5.6.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	34
5.6.3.3	GetNumber_of_alcatel()	35
5.6.3.4	GetNumber_of_huawei()	35
5.6.3.5	GetNumber_of_iphones()	35
5.6.3.6	GetNumber_of_nokia()	35
5.6.3.7	GetNumber_of_samsung()	35
5.6.3.8	GetNumber_of_sony()	35
5.6.3.9	GetShop_address()	36
5.6.3.10	GetShop_name()	36
5.6.3.11	operator=(const CARPHONE_WAREHOUSE &orig)	36
5.6.3.12	SetNumber_of_alcatel(int _number_of_alcatel)	36
5.6.3.13	SetNumber_of_huawei(int _number_of_huawei)	36
5.6.3.14	SetNumber_of_iphones(int _number_of_iphones)	36
5.6.3.15	SetNumber_of_nokia(int _number_of_nokia)	37

5.6.3.16	SetNumber_of_samsung(int _number_of_samsung)	37
5.6.3.17	SetNumber_of_sony(int _number_of_sony)	37
5.6.3.18	SetShop_address(std::string _shop_address)	37
5.6.3.19	SetShop_name(std::string _shop_name)	37
5.6.3.20	toString()	37
5.7	DUNDALK_W Class Reference	38
5.7.1	Detailed Description	38
5.7.2	Constructor & Destructor Documentation	39
5.7.2.1	DUNDALK_W()	39
5.7.2.2	DUNDALK_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	39
5.7.2.3	DUNDALK_W(std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id)	39
5.7.2.4	DUNDALK_W(const DUNDALK_W &orig)	39
5.7.2.5	~DUNDALK_W()	39
5.7.3	Member Function Documentation	39
5.7.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	39
5.7.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	40
5.7.3.3	GetNumber_of_alcatel()	40
5.7.3.4	GetNumber_of_huawei()	40
5.7.3.5	GetNumber_of_iphones()	40
5.7.3.6	GetNumber_of_nokia()	41
5.7.3.7	GetNumber_of_samsung()	41
5.7.3.8	GetNumber_of_sony()	41
5.7.3.9	GetShop_address()	41
5.7.3.10	GetShop_name()	41
5.7.3.11	operator=(const DUNDALK_W &orig)	41
5.7.3.12	SetNumber_of_alcatel(int _number_of_alcatel)	42
5.7.3.13	SetNumber_of_huawei(int _number_of_huawei)	42
5.7.3.14	SetNumber_of_iphones(int _number_of_iphones)	42
5.7.3.15	SetNumber_of_nokia(int _number_of_nokia)	42

5.7.3.16	SetNumber_of_samsung(int _number_of_samsung)	42
5.7.3.17	SetNumber_of_sony(int _number_of_sony)	42
5.7.3.18	SetShop_address(std::string _shop_address)	43
5.7.3.19	SetShop_name(std::string _shop_name)	43
5.7.3.20	toString()	43
5.8	KILKENNY_W Class Reference	43
5.8.1	Detailed Description	44
5.8.2	Constructor & Destructor Documentation	44
5.8.2.1	KILKENNY_W()	44
5.8.2.2	KILKENNY_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	45
5.8.2.3	KILKENNY_W(std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id)	45
5.8.2.4	KILKENNY_W(const KILKENNY_W &orig)	45
5.8.2.5	~KILKENNY_W()	45
5.8.3	Member Function Documentation	45
5.8.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	45
5.8.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	46
5.8.3.3	GetNumber_of_alcatel()	46
5.8.3.4	GetNumber_of_huawei()	46
5.8.3.5	GetNumber_of_iphones()	46
5.8.3.6	GetNumber_of_nokia()	46
5.8.3.7	GetNumber_of_samsung()	47
5.8.3.8	GetNumber_of_sony()	47
5.8.3.9	GetShop_address()	47
5.8.3.10	GetShop_name()	47
5.8.3.11	operator=(const KILKENNY_W &orig)	47
5.8.3.12	SetNumber_of_alcatel(int _number_of_alcatel)	47
5.8.3.13	SetNumber_of_huawei(int _number_of_huawei)	48
5.8.3.14	SetNumber_of_iphones(int _number_of_iphones)	48
5.8.3.15	SetNumber_of_nokia(int _number_of_nokia)	48

5.8.3.16	SetNumber_of_samsung(int _number_of_samsung)	48
5.8.3.17	SetNumber_of_sony(int _number_of_sony)	48
5.8.3.18	SetShop_address(std::string _shop_address)	48
5.8.3.19	SetShop_name(std::string _shop_name)	49
5.8.3.20	toString()	49
5.9	OSTM Class Reference	49
5.9.1	Detailed Description	50
5.9.2	Constructor & Destructor Documentation	50
5.9.2.1	OSTM()	50
5.9.2.2	OSTM(int _version_number_, int _unique_id_)	50
5.9.2.3	~OSTM()	51
5.9.3	Member Function Documentation	51
5.9.3.1	copy(std::shared_ptr< OSTM > from, std::shared_ptr< OSTM > to)	51
5.9.3.2	Get_Unique_ID() const	51
5.9.3.3	Get_Version() const	51
5.9.3.4	getBaseCopy(std::shared_ptr< OSTM > object)	52
5.9.3.5	increase_VersionNumber()	52
5.9.3.6	Is_Abort_Transaction() const	52
5.9.3.7	Is_Can_Commit() const	52
5.9.3.8	is_Locked()	53
5.9.3.9	lock_Mutex()	53
5.9.3.10	Set_Abort_Transaction(bool abortTransaction)	53
5.9.3.11	Set_Can_Commit(bool canCommit)	53
5.9.3.12	Set_Unique_ID(int uniqueID)	54
5.9.3.13	Set_Version(int version)	54
5.9.3.14	toString()	54
5.9.3.15	unlock_Mutex()	54
5.10	SLIGO_W Class Reference	55
5.10.1	Detailed Description	56
5.10.2	Constructor & Destructor Documentation	56

5.10.2.1	SLIGO_W()	56
5.10.2.2	SLIGO_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	56
5.10.2.3	SLIGO_W(std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id)	56
5.10.2.4	SLIGO_W(const SLIGO_W &orig)	56
5.10.2.5	~SLIGO_W()	56
5.10.3	Member Function Documentation	56
5.10.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	56
5.10.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	57
5.10.3.3	GetNumber_of_alcatel()	57
5.10.3.4	GetNumber_of_huawei()	57
5.10.3.5	GetNumber_of_iphones()	57
5.10.3.6	GetNumber_of_nokia()	58
5.10.3.7	GetNumber_of_samsung()	58
5.10.3.8	GetNumber_of_sony()	58
5.10.3.9	GetShop_address()	58
5.10.3.10	GetShop_name()	58
5.10.3.11	operator=(const SLIGO_W &orig)	58
5.10.3.12	SetNumber_of_alcatel(int _number_of_alcatel)	59
5.10.3.13	SetNumber_of_huawei(int _number_of_huawei)	59
5.10.3.14	SetNumber_of_iphones(int _number_of_iphones)	59
5.10.3.15	SetNumber_of_nokia(int _number_of_nokia)	59
5.10.3.16	SetNumber_of_samsung(int _number_of_samsung)	59
5.10.3.17	SetNumber_of_sony(int _number_of_sony)	59
5.10.3.18	SetShop_address(std::string _shop_address)	60
5.10.3.19	SetShop_name(std::string _shop_name)	60
5.10.3.20	toString()	60
5.11	SWBPLC Class Reference	60
5.11.1	Detailed Description	61
5.11.2	Constructor & Destructor Documentation	61
5.11.2.1	SWBPLC()	61

5.11.2.2	SWBPLC(int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)	61
5.11.2.3	SWBPLC(std::shared_ptr< BANK > obj, int _version, int _unique_id)	62
5.11.2.4	SWBPLC(const SWBPLC &orig)	62
5.11.2.5	~SWBPLC()	62
5.11.3	Member Function Documentation	62
5.11.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	62
5.11.3.2	GetAccountNumber() const	62
5.11.3.3	GetAddress() const	63
5.11.3.4	GetBalance() const	63
5.11.3.5	getBaseCopy(std::shared_ptr< OSTM > object)	63
5.11.3.6	GetFirstName() const	63
5.11.3.7	GetFullname() const	63
5.11.3.8	GetLastName() const	64
5.11.3.9	operator=(const SWBPLC &orig)	64
5.11.3.10	SetAccountNumber(int accountNumber)	64
5.11.3.11	SetAddress(std::string address)	64
5.11.3.12	SetBalance(double balance)	64
5.11.3.13	SetFirstName(std::string firstName)	64
5.11.3.14	SetFullname(std::string fullname)	65
5.11.3.15	SetLastName(std::string lastName)	65
5.11.3.16	toString()	65
5.12	TALLAGH_W Class Reference	65
5.12.1	Detailed Description	66
5.12.2	Constructor & Destructor Documentation	66
5.12.2.1	TALLAGH_W()	66
5.12.2.2	TALLAGH_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	67
5.12.2.3	TALLAGH_W(std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id)	67
5.12.2.4	TALLAGH_W(const TALLAGH_W &orig)	67
5.12.2.5	~TALLAGH_W()	67

5.12.3	Member Function Documentation	67
5.12.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	67
5.12.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	68
5.12.3.3	GetNumber_of_alcatel()	68
5.12.3.4	GetNumber_of_huawei()	68
5.12.3.5	GetNumber_of_iphones()	68
5.12.3.6	GetNumber_of_nokia()	68
5.12.3.7	GetNumber_of_samsung()	69
5.12.3.8	GetNumber_of_sony()	69
5.12.3.9	GetShop_address()	69
5.12.3.10	GetShop_name()	69
5.12.3.11	operator=(const TALLAGH_W &orig)	69
5.12.3.12	SetNumber_of_alcatel(int _number_of_alcatel)	69
5.12.3.13	SetNumber_of_huawei(int _number_of_huawei)	70
5.12.3.14	SetNumber_of_iphones(int _number_of_iphones)	70
5.12.3.15	SetNumber_of_nokia(int _number_of_nokia)	70
5.12.3.16	SetNumber_of_samsung(int _number_of_samsung)	70
5.12.3.17	SetNumber_of_sony(int _number_of_sony)	70
5.12.3.18	SetShop_address(std::string _shop_address)	70
5.12.3.19	SetShop_name(std::string _shop_name)	71
5.12.3.20	toString()	71
5.13	TM Class Reference	71
5.13.1	Detailed Description	71
5.13.2	Member Function Documentation	71
5.13.2.1	_get_tx()	71
5.13.2.2	_TX_EXIT()	72
5.13.2.3	Instance()	72
5.13.2.4	print_all()	72
5.14	TX Class Reference	73
5.14.1	Detailed Description	73

5.14.2	Constructor & Destructor Documentation	73
5.14.2.1	TX(std::thread::id id)	73
5.14.2.2	~TX()	74
5.14.2.3	TX(const TX &orig)	74
5.14.3	Member Function Documentation	74
5.14.3.1	_decrease_tx_nesting()	74
5.14.3.2	_increase_tx_nesting()	74
5.14.3.3	_print_all_tx()	75
5.14.3.4	_register(std::shared_ptr< OSTM > object)	75
5.14.3.5	commit()	75
5.14.3.6	getTest_counter()	75
5.14.3.7	load(std::shared_ptr< OSTM > object)	76
5.14.3.8	ostm_exit()	76
5.14.3.9	store(std::shared_ptr< OSTM > object)	76
5.14.4	Friends And Related Function Documentation	76
5.14.4.1	TM	76
5.14.5	Member Data Documentation	77
5.14.5.1	test_counter	77
5.15	ULSTER Class Reference	77
5.15.1	Detailed Description	78
5.15.2	Constructor & Destructor Documentation	78
5.15.2.1	ULSTER()	78
5.15.2.2	ULSTER(int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)	78
5.15.2.3	ULSTER(std::shared_ptr< BANK > obj, int _version, int _unique_id)	78
5.15.2.4	ULSTER(const ULSTER &orig)	78
5.15.2.5	~ULSTER()	78
5.15.3	Member Function Documentation	78
5.15.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	78
5.15.3.2	GetAccountNumber() const	79
5.15.3.3	GetAddress() const	79

5.15.3.4	GetBalance() const	79
5.15.3.5	getBaseCopy(std::shared_ptr< OSTM > object)	79
5.15.3.6	GetFirstName() const	80
5.15.3.7	GetFullname() const	80
5.15.3.8	GetLastName() const	80
5.15.3.9	operator=(const ULSTER &orig)	80
5.15.3.10	SetAccountNumber(int accountNumber)	80
5.15.3.11	SetAddress(std::string address)	81
5.15.3.12	SetBalance(double balance)	81
5.15.3.13	SetFirstName(std::string firstName)	81
5.15.3.14	SetFullname(std::string fullname)	81
5.15.3.15	SetLastName(std::string lastName)	81
5.15.3.16	toString()	81
5.16	UNBL Class Reference	82
5.16.1	Detailed Description	82
5.16.2	Constructor & Destructor Documentation	82
5.16.2.1	UNBL()	82
5.16.2.2	UNBL(int accountNumber, double balance, std::string firstName, std::string last↵ Name, std::string address)	83
5.16.2.3	UNBL(std::shared_ptr< BANK > obj, int _version, int _unique_id)	83
5.16.2.4	UNBL(const UNBL &orig)	83
5.16.2.5	~UNBL()	83
5.16.3	Member Function Documentation	83
5.16.3.1	copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)	83
5.16.3.2	GetAccountNumber() const	84
5.16.3.3	GetAddress() const	84
5.16.3.4	GetBalance() const	84
5.16.3.5	getBaseCopy(std::shared_ptr< OSTM > object)	84
5.16.3.6	GetFirstName() const	84
5.16.3.7	GetFullname() const	85
5.16.3.8	GetLastName() const	85

5.16.3.9	operator=(const UNBL &orig)	85
5.16.3.10	SetAccountNumber(int accountNumber)	85
5.16.3.11	SetAddress(std::string address)	85
5.16.3.12	SetBalance(double balance)	85
5.16.3.13	SetFirstName(std::string firstName)	86
5.16.3.14	SetFullname(std::string fullname)	86
5.16.3.15	SetLastName(std::string lastName)	86
5.16.3.16	toString()	86
5.17	WAREHOUSE Class Reference	86
5.17.1	Detailed Description	87
5.17.2	Constructor & Destructor Documentation	87
5.17.2.1	WAREHOUSE()	87
5.17.2.2	WAREHOUSE(int _version, int _unique_id)	87
5.17.2.3	WAREHOUSE(const WAREHOUSE &orig)	88
5.17.2.4	~WAREHOUSE()	88
5.17.3	Member Function Documentation	88
5.17.3.1	GetNumber_of_alcatel()	88
5.17.3.2	GetNumber_of_huawei()	88
5.17.3.3	GetNumber_of_iphones()	88
5.17.3.4	GetNumber_of_nokia()	88
5.17.3.5	GetNumber_of_samsung()	89
5.17.3.6	GetNumber_of_sony()	89
5.17.3.7	GetShop_address()	89
5.17.3.8	GetShop_name()	89
5.17.3.9	SetNumber_of_alcatel(int _number_of_alcatel)	89
5.17.3.10	SetNumber_of_huawei(int _number_of_huawei)	89
5.17.3.11	SetNumber_of_iphones(int _number_of_iphones)	89
5.17.3.12	SetNumber_of_nokia(int _number_of_nokia)	90
5.17.3.13	SetNumber_of_samsung(int _number_of_samsung)	90
5.17.3.14	SetNumber_of_sony(int _number_of_sony)	90
5.17.3.15	SetShop_address(std::string _shop_address)	90
5.17.3.16	SetShop_name(std::string _shop_name)	90

6 File Documentation	91
6.1 AIB.cpp File Reference	91
6.2 AIB.h File Reference	91
6.3 BANK.cpp File Reference	91
6.4 BANK.h File Reference	91
6.5 BOA.cpp File Reference	92
6.6 BOA.h File Reference	92
6.7 BOI.cpp File Reference	92
6.8 BOI.h File Reference	92
6.9 CARLOW_W.cpp File Reference	92
6.10 CARLOW_W.h File Reference	93
6.11 CARPHONE_WAREHOUSE.cpp File Reference	93
6.12 CARPHONE_WAREHOUSE.h File Reference	93
6.13 DUNDALK_W.cpp File Reference	93
6.14 DUNDALK_W.h File Reference	93
6.15 KILKENNY_W.cpp File Reference	94
6.16 KILKENNY_W.h File Reference	94
6.17 main.cpp File Reference	94
6.17.1 Function Documentation	95
6.17.1.1 _complex_transfer_(std::shared_ptr< OSTM > _from_, std::shared_ptr< OSTM > _from_two_, std::vector< std::shared_ptr< OSTM >> _customer_vec, TM &_tm, double _amount)	95
6.17.1.2 _complex_warehouse_transfer_(std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _to_two_, std::shared_ptr< OSTM > _to_three_, std::vector< std::shared_ptr< OSTM >> _warehouse_vec, std::shared_ptr< OSTM > _from_, TM &_tm, double _amount)	96
6.17.1.3 _nested_warehouse_transfer_(std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _to_two_, std::shared_ptr< OSTM > _to_three_, std::shared_ptr< OSTM > _from_, TM &_tm, double _amount)	96
6.17.1.4 _nesting_(std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM &_tm, double _amount)	97
6.17.1.5 _six_account_transfer_(std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_one_, std::shared_ptr< OSTM > _from_two_, std::shared_ptr< OSTM > _from_three_, std::shared_ptr< OSTM > _from_four_, std::shared_ptr< OSTM > _from_five_, TM &_tm, double _amount)	98

6.17.1.6	<code>_two_account_transfer_(std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM & _tm, double _amount)</code>	99
6.17.1.7	<code>_warehouse_transfer_(std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM & _tm, double _amount)</code>	99
6.17.1.8	<code>main(void)</code>	100
6.18	OSTM.cpp File Reference	102
6.19	OSTM.h File Reference	102
6.20	SLIGO_W.cpp File Reference	102
6.21	SLIGO_W.h File Reference	102
6.22	SWBPLC.cpp File Reference	103
6.23	SWBPLC.h File Reference	103
6.24	TALLAGH_W.cpp File Reference	103
6.25	TALLAGH_W.h File Reference	103
6.26	TM.cpp File Reference	103
6.27	TM.h File Reference	104
6.28	TX.cpp File Reference	104
6.29	TX.h File Reference	104
6.30	ULSTER.cpp File Reference	104
6.31	ULSTER.h File Reference	105
6.32	UNBL.cpp File Reference	105
6.33	UNBL.h File Reference	105
6.34	WAREHOUSE.cpp File Reference	105
6.35	WAREHOUSE.h File Reference	105

Chapter 1

OSTM C++ Software Transactional Memory

1.1 Object Based Software Transactional Memory.

OSTM is a polymorphic solution to store and manage shared memory spaces within c++ programming context. You can store and managed any kind of object in transactional environment as a shared and protected memory space.

1.1.1 Brief. Download the zip file from the provided link in the web-site, that contains the libostm.so, TM.h, TX.h, OSTM.h files.

Unzip the archive file to the desired destination possibly where in you program is stored.

1.1.2 Step 1: Download the archive file.

1.1.3 Step 2: Unzip in the target destination.

1.1.4 Step 3: Copy the shared library (libostm.so) to the operating system folder where the other shared library are stored.

It will be different destination folder on different platforms. (Linux, Windows, Mac OS) [More Information](#)

1.1.5 Step 4: Achieve the required class hierarchy between the OSTM library and your own class structure.

Details and instruction of class hierarchy requirements can be found on the web-site. www.serversite.info/ostm

1.1.6 Step 5: Create an executable file as you linking together the TM.h, TX.h, OSTM.h files with your own files.

1.1.7 Step 6: Now your application use transactional environment, that guarantees the consistency between object transactions.

1.1.8 Step 7: Run the application.

Abbreviation for bank names used in the test cases:

BOA - Bank of America

ULSTER - Ulster Bank

UNBL - United National Bank Limited

SWBPLC - Scottish Windows Bank PLC

AIB - Allied Irish Bank

BOI - Bank of Ireland

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

OSTM	49
BANK	14
AIB	9
BOA	17
BOI	21
SWBPLC	60
ULSTER	77
UNBL	82
WAREHOUSE	86
CARLOW_W	26
CARPHONE_WAREHOUSE	32
DUNDALK_W	38
KILKENNY_W	43
SLIGO_W	55
TALLAGH_W	65
TM	71
TX	73

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AIB	9
BANK	14
BOA	17
BOI	21
CARLOW_W	26
CARPHONE_WAREHOUSE	32
DUNDALK_W	38
KILKENNY_W	43
OSTM	49
SLIGO_W	55
SWBPLC	60
TALLAGH_W	65
TM	71
TX	73
ULSTER	77
UNBL	82
WAREHOUSE	86

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

AIB.cpp	91
AIB.h	91
BANK.cpp	91
BANK.h	91
BOA.cpp	92
BOA.h	92
BOI.cpp	92
BOI.h	92
CARLOW_W.cpp	92
CARLOW_W.h	93
CARPHONE_WAREHOUSE.cpp	93
CARPHONE_WAREHOUSE.h	93
DUNDALK_W.cpp	93
DUNDALK_W.h	93
KILKENNY_W.cpp	94
KILKENNY_W.h	94
main.cpp	94
OSTM.cpp	102
OSTM.h	102
SLIGO_W.cpp	102
SLIGO_W.h	102
SWBPLC.cpp	103
SWBPLC.h	103
TALLAGH_W.cpp	103
TALLAGH_W.h	103
TM.cpp	103
TM.h	104
TX.cpp	104
TX.h	104
ULSTER.cpp	104
ULSTER.h	105
UNBL.cpp	105
UNBL.h	105
WAREHOUSE.cpp	105
WAREHOUSE.h	105

Chapter 5

Class Documentation

5.1 AIB Class Reference

```
#include <AIB.h>
```

Inheritance diagram for AIB:

Collaboration diagram for AIB:

Public Member Functions

- [AIB](#) ()
- [AIB](#) (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- [AIB](#) (std::shared_ptr< [BANK](#) > obj, int _version, int _unique_id)
- [AIB](#) (const [AIB](#) &orig)
- [AIB operator=](#) (const [AIB](#) &orig)
- virtual [~AIB](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetAddress](#) (std::string address)
- virtual std::string [GetAddress](#) () const
- virtual void [SetBalance](#) (double balance)
- virtual double [GetBalance](#) () const
- virtual void [SetAccountNumber](#) (int accountNumber)
- virtual int [GetAccountNumber](#) () const
- virtual void [SetLastName](#) (std::string lastName)
- virtual std::string [GetLastName](#) () const
- virtual void [SetFirstName](#) (std::string firstName)
- virtual std::string [GetFirstName](#) () const
- virtual void [SetFullname](#) (std::string fullname)
- virtual std::string [GetFullname](#) () const

5.1.1 Detailed Description

Inherit from [BANK](#)

Definition at line 18 of file AIB.h.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 AIB::AIB () [inline]

Constructor

Definition at line 23 of file AIB.h.

Here is the caller graph for this function:

5.1.2.2 AIB::AIB (int *accountNumber*, double *balance*, std::string *firstName*, std::string *lastName*, std::string *address*) [inline]

Custom constructor

Definition at line 36 of file AIB.h.

5.1.2.3 AIB::AIB (std::shared_ptr< BANK > *obj*, int *_version*, int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 48 of file AIB.h.

Here is the call graph for this function:

5.1.2.4 AIB::AIB (const AIB & *orig*)

Copy constructor

Definition at line 14 of file AIB.cpp.

5.1.2.5 AIB::~AIB () [virtual]

de-constructor

Definition at line 17 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3 Member Function Documentation

5.1.3.1 void AIB::copy (std::shared_ptr< OSTM > *to*, std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a <code>std::shared_ptr<BANK></code> type object casted back from <code>std::shared_ptr<OSTM></code>
<i>objFROM</i>	is a <code>std::shared_ptr<BANK></code> type object casted back from <code>std::shared_ptr<OSTM></code>

Reimplemented from [OSTM](#).

Definition at line 37 of file AIB.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.1.3.2 `int AIB::GetAccountNumber () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 81 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.3 `std::string AIB::GetAddress () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 65 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.4 `double AIB::GetBalance () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 73 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.5 `std::shared_ptr<OSTM> AIB::getBaseCopy (std::shared_ptr<OSTM> object)` `[virtual]`

getBaseCopy function, make deep copy of the object/pointer and Return a new `std::shared_ptr<BANK>` type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a <code>std::shared_ptr<BANK></code> return type

Reimplemented from [OSTM](#).

Definition at line 24 of file AIB.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.1.3.6 `std::string AIB::GetFirstName () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 97 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.7 `std::string AIB::GetFullname () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 105 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.8 `std::string AIB::GetLastName () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 89 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.9 `AIB AIB::operator= (const AIB & orig)` `[inline]`

Operator

Definition at line 66 of file AIB.h.

Here is the call graph for this function:

5.1.3.10 `void AIB::SetAccountNumber (int accountNumber)` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 77 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.11 `void AIB::SetAddress (std::string address) [virtual]`

Reimplemented from [BANK](#).

Definition at line 61 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.12 `void AIB::SetBalance (double balance) [virtual]`

Reimplemented from [BANK](#).

Definition at line 69 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.13 `void AIB::SetFirstName (std::string firstName) [virtual]`

Reimplemented from [BANK](#).

Definition at line 93 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.14 `void AIB::SetFullname (std::string fullname) [virtual]`

Reimplemented from [BANK](#).

Definition at line 101 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.15 `void AIB::SetLastName (std::string lastName) [virtual]`

Reimplemented from [BANK](#).

Definition at line 85 of file AIB.cpp.

Here is the caller graph for this function:

5.1.3.16 `void AIB::toString () [virtual]`

`_cast`, is use to cast bak the `std::shared_ptr<OSTM>` to the required type

`toString` function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 56 of file AIB.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [AIB.h](#)
- [AIB.cpp](#)

5.2 BANK Class Reference

```
#include <BANK.h>
```

Inheritance diagram for BANK:

Collaboration diagram for BANK:

Public Member Functions

- [BANK](#) ()
- [BANK](#) (int _version, int _unique_id)
- [BANK](#) (const [BANK](#) &orig)
- virtual [~BANK](#) ()
- virtual void [SetAddress](#) (std::string address)
- virtual std::string [GetAddress](#) () const
- virtual void [SetBalance](#) (double balance)
- virtual double [GetBalance](#) () const
- virtual void [SetAccountNumber](#) (int accountNumber)
- virtual int [GetAccountNumber](#) () const
- virtual void [SetLastName](#) (std::string lastName)
- virtual std::string [GetLastName](#) () const
- virtual void [SetFirstName](#) (std::string firstName)
- virtual std::string [GetFirstName](#) () const
- virtual void [SetFullname](#) (std::string fullname)
- virtual std::string [GetFullname](#) () const

5.2.1 Detailed Description

[BANK](#) inherit from the [OSTM](#) library. It is declares the common functions in the child classes as a virtual function.

Definition at line 16 of file BANK.h.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 [BANK::BANK \(\)](#) `[inline]`

Constructor

Definition at line 23 of file BANK.h.

Here is the caller graph for this function:

5.2.2.2 [BANK::BANK \(int _version, int _unique_id \)](#) `[inline]`

Custom Constructor

Definition at line 29 of file BANK.h.

Here is the call graph for this function:

5.2.2.3 BANK::BANK (const BANK & orig)

Copy constructor

Definition at line 11 of file BANK.cpp.

5.2.2.4 BANK::~~BANK () [virtual]

de-constructor

Definition at line 14 of file BANK.cpp.

Here is the caller graph for this function:

5.2.3 Member Function Documentation

5.2.3.1 virtual int BANK::GetAccountNumber () const [inline],[virtual]

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 49 of file BANK.h.

5.2.3.2 virtual std::string BANK::GetAddress () const [inline],[virtual]

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 45 of file BANK.h.

5.2.3.3 virtual double BANK::GetBalance () const [inline],[virtual]

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 47 of file BANK.h.

5.2.3.4 virtual std::string BANK::GetFirstName () const [inline],[virtual]

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 53 of file BANK.h.

5.2.3.5 virtual std::string BANK::GetFullname () const [inline],[virtual]

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 55 of file BANK.h.

5.2.3.6 `virtual std::string BANK::GetLastName () const [inline],[virtual]`

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 51 of file BANK.h.

5.2.3.7 `virtual void BANK::SetAccountNumber (int accountNumber) [inline],[virtual]`

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 48 of file BANK.h.

5.2.3.8 `virtual void BANK::SetAddress (std::string address) [inline],[virtual]`

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 44 of file BANK.h.

5.2.3.9 `virtual void BANK::SetBalance (double balance) [inline],[virtual]`

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 46 of file BANK.h.

Here is the caller graph for this function:

5.2.3.10 `virtual void BANK::SetFirstName (std::string firstName) [inline],[virtual]`

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 52 of file BANK.h.

5.2.3.11 `virtual void BANK::SetFullname (std::string fullname) [inline],[virtual]`

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 54 of file BANK.h.

5.2.3.12 `virtual void BANK::SetLastName (std::string lastName) [inline],[virtual]`

Reimplemented in [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 50 of file BANK.h.

The documentation for this class was generated from the following files:

- [BANK.h](#)
- [BANK.cpp](#)

5.3 BOA Class Reference

```
#include <BOA.h>
```

Inheritance diagram for BOA:

Collaboration diagram for BOA:

Public Member Functions

- [BOA](#) ()
- [BOA](#) (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- [BOA](#) (std::shared_ptr< [BANK](#) > obj, int _version, int _unique_id)
- [BOA](#) (const [BOA](#) &orig)
- [BOA operator=](#) (const [BOA](#) &orig)
- virtual [~BOA](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetAddress](#) (std::string address)
- virtual std::string [GetAddress](#) () const
- virtual void [SetBalance](#) (double balance)
- virtual double [GetBalance](#) () const
- virtual void [SetAccountNumber](#) (int accountNumber)
- virtual int [GetAccountNumber](#) () const
- virtual void [SetLastName](#) (std::string lastName)
- virtual std::string [GetLastName](#) () const
- virtual void [SetFirstName](#) (std::string firstName)
- virtual std::string [GetFirstName](#) () const
- virtual void [SetFullname](#) (std::string fullname)
- virtual std::string [GetFullname](#) () const

5.3.1 Detailed Description

Inherit from [BANK](#)

Definition at line 18 of file BOA.h.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 [BOA::BOA](#) () `[inline]`

Constructor

Definition at line 24 of file BOA.h.

Here is the caller graph for this function:

5.3.2.2 `BOA::BOA (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)`
`[inline]`

Custom constructor

Definition at line 35 of file BOA.h.

5.3.2.3 `BOA::BOA (std::shared_ptr< BANK > obj, int _version, int _unique_id)` `[inline]`

Custom constructor, used by the library for deep copying

Definition at line 46 of file BOA.h.

Here is the call graph for this function:

5.3.2.4 `BOA::BOA (const BOA & orig)`

Copy constructor

Definition at line 12 of file BOA.cpp.

5.3.2.5 `BOA::~BOA ()` `[virtual]`

de-constructor

Definition at line 15 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3 Member Function Documentation

5.3.3.1 `void BOA::copy (std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)` `[virtual]`

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 34 of file BOA.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.3.3.2 `int BOA::GetAccountNumber () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 80 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.3 `std::string BOA::GetAddress () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 64 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.4 `double BOA::GetBalance () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 72 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.5 `std::shared_ptr< OSTM > BOA::getBaseCopy (std::shared_ptr< OSTM > object) [virtual]`

getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a std::shared_ptr<BANK> return type

Reimplemented from [OSTM](#).

Definition at line 22 of file BOA.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.3.3.6 `std::string BOA::GetFirstName () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 96 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.7 `std::string BOA::GetFullName () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 104 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.8 `std::string BOA::GetLastName () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 88 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.9 `BOA BOA::operator= (const BOA & orig)` [inline]

Operator

Definition at line 64 of file BOA.h.

Here is the call graph for this function:

5.3.3.10 `void BOA::SetAccountNumber (int accountNumber)` [virtual]

Reimplemented from [BANK](#).

Definition at line 76 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.11 `void BOA::SetAddress (std::string address)` [virtual]

Reimplemented from [BANK](#).

Definition at line 60 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.12 `void BOA::SetBalance (double balance)` [virtual]

Reimplemented from [BANK](#).

Definition at line 68 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.13 `void BOA::SetFirstName (std::string firstName)` [virtual]

Reimplemented from [BANK](#).

Definition at line 92 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.14 `void BOA::SetFullname (std::string fullname)` [virtual]

Reimplemented from [BANK](#).

Definition at line 100 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.15 `void BOA::SetLastName (std::string lastName)` [virtual]

Reimplemented from [BANK](#).

Definition at line 84 of file BOA.cpp.

Here is the caller graph for this function:

5.3.3.16 `void BOA::toString ()` [virtual]

`_cast`, is use to cast bak the `std::shared_ptr<OSTM>` to the required type

`toString` function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 54 of file BOA.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [BOA.h](#)
- [BOA.cpp](#)

5.4 BOI Class Reference

```
#include <BOI.h>
```

Inheritance diagram for BOI:

Collaboration diagram for BOI:

Public Member Functions

- [BOI](#) ()
- [BOI](#) (int *accountNumber*, double *balance*, std::string *firstName*, std::string *lastName*, std::string *address*)
- [BOI](#) (std::shared_ptr< [BOI](#) > *obj*, int *_version*, int *_unique_id*)
- [BOI](#) (const [BOI](#) &*orig*)
- [BOI operator=](#) (const [BOI](#) &*orig*)
- virtual [~BOI](#) ()
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > *object*)
getBaseCopy function, make deep copy of the object/pointer and Return a new BANK type object*
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > *to*, std::shared_ptr< [OSTM](#) > *from*)
copy function, make deep copy of the object/pointer
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetAddress](#) (std::string *address*)
- virtual std::string [GetAddress](#) () const
- virtual void [SetBalance](#) (double *balance*)
- virtual double [GetBalance](#) () const
- virtual void [SetAccountNumber](#) (int *accountNumber*)
- virtual int [GetAccountNumber](#) () const
- virtual void [SetLastName](#) (std::string *lastName*)
- virtual std::string [GetLastName](#) () const
- virtual void [SetFirstName](#) (std::string *firstName*)
- virtual std::string [GetFirstName](#) () const
- virtual void [SetFullname](#) (std::string *fullname*)
- virtual std::string [GetFullname](#) () const

5.4.1 Detailed Description

Inherit from [BANK](#)

Definition at line 19 of file [BOI.h](#).

5.4.2 Constructor & Destructor Documentation

5.4.2.1 [BOI::BOI](#) () `[inline]`

Constructor

Definition at line 24 of file [BOI.h](#).

Here is the caller graph for this function:

5.4.2.2 [BOI::BOI](#) (int *accountNumber*, double *balance*, std::string *firstName*, std::string *lastName*, std::string *address*) `[inline]`

Custom constructor

Definition at line 37 of file [BOI.h](#).

5.4.2.3 `BOI::BOI (std::shared_ptr< BOI > obj, int_version, int_unique_id) [inline]`

Custom constructor, used by the library for deep copying

Definition at line 49 of file BOI.h.

Here is the call graph for this function:

5.4.2.4 `BOI::BOI (const BOI & orig)`

Copy constructor

Definition at line 15 of file BOI.cpp.

5.4.2.5 `BOI::~BOI () [virtual]`

de-constructor

Definition at line 12 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3 Member Function Documentation

5.4.3.1 `void BOI::copy (std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) [virtual]`

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 35 of file BOI.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.4.3.2 `int BOI::GetAccountNumber () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 78 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3.3 `std::string BOl::GetAddress () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 62 of file BOl.cpp.

Here is the caller graph for this function:

5.4.3.4 `double BOl::GetBalance () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 70 of file BOl.cpp.

Here is the caller graph for this function:

5.4.3.5 `std::shared_ptr< OSTM > BOl::getBaseCopy (std::shared_ptr< OSTM > object)` [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK* type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a BANK* return type

Reimplemented from [OSTM](#).

Definition at line 22 of file BOl.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.4.3.6 `std::string BOl::GetFirstName () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 94 of file BOl.cpp.

Here is the caller graph for this function:

5.4.3.7 `std::string BOl::GetFullname () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 102 of file BOl.cpp.

Here is the caller graph for this function:

5.4.3.8 `std::string BOI::GetLastName () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 86 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3.9 `BOI BOI::operator= (const BOI & orig) [inline]`

Operator

Definition at line 65 of file BOI.h.

Here is the call graph for this function:

5.4.3.10 `void BOI::SetAccountNumber (int accountNumber) [virtual]`

Reimplemented from [BANK](#).

Definition at line 74 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3.11 `void BOI::SetAddress (std::string address) [virtual]`

Reimplemented from [BANK](#).

Definition at line 58 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3.12 `void BOI::SetBalance (double balance) [virtual]`

Reimplemented from [BANK](#).

Definition at line 66 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3.13 `void BOI::SetFirstName (std::string firstName) [virtual]`

Reimplemented from [BANK](#).

Definition at line 90 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3.14 `void BOI::SetFullname (std::string fullname)` [virtual]

Reimplemented from [BANK](#).

Definition at line 98 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3.15 `void BOI::SetLastName (std::string lastName)` [virtual]

Reimplemented from [BANK](#).

Definition at line 82 of file BOI.cpp.

Here is the caller graph for this function:

5.4.3.16 `void BOI::toString ()` [virtual]

`_cast`, is use to cast bak the `std::shared_ptr<OSTM>` to the required type

`toString` function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 54 of file BOI.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [BOI.h](#)
- [BOI.cpp](#)

5.5 CARLOW_W Class Reference

```
#include <CARLOW_W.h>
```

Inheritance diagram for CARLOW_W:

Collaboration diagram for CARLOW_W:

Public Member Functions

- [CARLOW_W](#) ()
- [CARLOW_W](#) (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- [CARLOW_W](#) (std::shared_ptr< [WAREHOUSE](#) > obj, int _version, int _unique_id)
- [CARLOW_W](#) (const [CARLOW_W](#) &orig)
- [CARLOW_W operator=](#) (const [CARLOW_W](#) &orig)
- virtual [~CARLOW_W](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new BANK type object*
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetNumber_of_alcatel](#) (int _number_of_alcatel)
- virtual int [GetNumber_of_alcatel](#) ()
- virtual void [SetNumber_of_nokia](#) (int _number_of_nokia)
- virtual int [GetNumber_of_nokia](#) ()
- virtual void [SetNumber_of_huawei](#) (int _number_of_huawei)
- virtual int [GetNumber_of_huawei](#) ()
- virtual void [SetNumber_of_sony](#) (int _number_of_sony)
- virtual int [GetNumber_of_sony](#) ()
- virtual void [SetNumber_of_samsung](#) (int _number_of_samsung)
- virtual int [GetNumber_of_samsung](#) ()
- virtual void [SetNumber_of_iphones](#) (int _number_of_iphones)
- virtual int [GetNumber_of_iphones](#) ()
- virtual void [SetShop_name](#) (std::string _shop_name)
- virtual std::string [GetShop_name](#) ()
- virtual void [SetShop_address](#) (std::string _shop_address)
- virtual std::string [GetShop_address](#) ()

5.5.1 Detailed Description

Inherit from [WAREHOUSE](#)

Definition at line 19 of file CARLOW_W.h.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 CARLOW_W::CARLOW_W () [inline]

Constructor

Definition at line 24 of file CARLOW_W.h.

Here is the caller graph for this function:

5.5.2.2 `CARLOW_W::CARLOW_W (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel) [inline]`

Custom constructor

Definition at line 38 of file CARLOW_W.h.

5.5.2.3 `CARLOW_W::CARLOW_W (std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id) [inline]`

Custom constructor, used by the library for deep copying

Definition at line 55 of file CARLOW_W.h.

Here is the call graph for this function:

5.5.2.4 `CARLOW_W::CARLOW_W (const CARLOW_W & orig)`

Copy constructor

Definition at line 17 of file CARLOW_W.cpp.

5.5.2.5 `CARLOW_W::~~CARLOW_W () [virtual]`

de-constructor

Definition at line 14 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3 Member Function Documentation

5.5.3.1 `void CARLOW_W::copy (std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) [virtual]`

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 37 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.2 `std::shared_ptr< OSTM > CARLOW_W::getBaseCopy (std::shared_ptr< OSTM > object)` [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK* type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a BANK* return type

Reimplemented from [OSTM](#).

Definition at line 24 of file CARLOW_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.5.3.3 `int CARLOW_W::GetNumber_of_alcatel ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 75 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.4 `int CARLOW_W::GetNumber_of_huawei ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 91 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.5 `int CARLOW_W::GetNumber_of_iphones ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 115 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.6 `int CARLOW_W::GetNumber_of_nokia ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 83 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.7 `int CARLOW_W::GetNumber_of_samsung () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 107 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.8 `int CARLOW_W::GetNumber_of_sony () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 99 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.9 `std::string CARLOW_W::GetShop_address () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 131 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.10 `std::string CARLOW_W::GetShop_name () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 123 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.11 `CARLOW_W CARLOW_W::operator= (const CARLOW_W & orig) [inline]`

Operator

Definition at line 75 of file CARLOW_W.h.

Here is the call graph for this function:

5.5.3.12 `void CARLOW_W::SetNumber_of_alcatel (int _number_of_alcatel) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 71 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.13 `void CARLOW_W::SetNumber_of_huawei (int number_of_huawei) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 87 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.14 `void CARLOW_W::SetNumber_of_iphones (int number_of_iphones) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 111 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.15 `void CARLOW_W::SetNumber_of_nokia (int number_of_nokia) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 79 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.16 `void CARLOW_W::SetNumber_of_samsung (int number_of_samsung) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 103 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.17 `void CARLOW_W::SetNumber_of_sony (int number_of_sony) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 95 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.18 `void CARLOW_W::SetShop_address (std::string shop_address) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 127 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.19 void CARLOW_W::SetShop_name (std::string_shop_name) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 119 of file CARLOW_W.cpp.

Here is the caller graph for this function:

5.5.3.20 void CARLOW_W::toString () [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 64 of file CARLOW_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [CARLOW_W.h](#)
- [CARLOW_W.cpp](#)

5.6 CARPHONE_WAREHOUSE Class Reference

```
#include <CARPHONE_WAREHOUSE.h>
```

Inheritance diagram for CARPHONE_WAREHOUSE:

Collaboration diagram for CARPHONE_WAREHOUSE:

Public Member Functions

- [CARPHONE_WAREHOUSE](#) ()
- [CARPHONE_WAREHOUSE](#) (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- [CARPHONE_WAREHOUSE](#) (std::shared_ptr< [WAREHOUSE](#) > obj, int _version, int _unique_id)
- [CARPHONE_WAREHOUSE](#) (const [CARPHONE_WAREHOUSE](#) &orig)
- [CARPHONE_WAREHOUSE](#) operator= (const [CARPHONE_WAREHOUSE](#) &orig)
- virtual ~[CARPHONE_WAREHOUSE](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new BANK type object*
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

- virtual void [SetNumber_of_alcatel](#) (int _number_of_alcatel)
- virtual int [GetNumber_of_alcatel](#) ()
- virtual void [SetNumber_of_nokia](#) (int _number_of_nokia)
- virtual int [GetNumber_of_nokia](#) ()
- virtual void [SetNumber_of_huawei](#) (int _number_of_huawei)
- virtual int [GetNumber_of_huawei](#) ()
- virtual void [SetNumber_of_sony](#) (int _number_of_sony)
- virtual int [GetNumber_of_sony](#) ()
- virtual void [SetNumber_of_samsung](#) (int _number_of_samsung)
- virtual int [GetNumber_of_samsung](#) ()
- virtual void [SetNumber_of_iphones](#) (int _number_of_iphones)
- virtual int [GetNumber_of_iphones](#) ()
- virtual void [SetShop_name](#) (std::string _shop_name)
- virtual std::string [GetShop_name](#) ()
- virtual void [SetShop_address](#) (std::string _shop_address)
- virtual std::string [GetShop_address](#) ()

5.6.1 Detailed Description

Inherit from [WAREHOUSE](#)

Definition at line 19 of file CARPHONE_WAREHOUSE.h.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 CARPHONE_WAREHOUSE::CARPHONE_WAREHOUSE () `[inline]`

Constructor

Definition at line 24 of file CARPHONE_WAREHOUSE.h.

Here is the caller graph for this function:

5.6.2.2 CARPHONE_WAREHOUSE::CARPHONE_WAREHOUSE (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel) `[inline]`

Custom constructor

Definition at line 38 of file CARPHONE_WAREHOUSE.h.

5.6.2.3 CARPHONE_WAREHOUSE::CARPHONE_WAREHOUSE (std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id) `[inline]`

Custom constructor, used by the library for deep copying

Definition at line 55 of file CARPHONE_WAREHOUSE.h.

Here is the call graph for this function:

5.6.2.4 CARPHONE_WAREHOUSE::CARPHONE_WAREHOUSE (const CARPHONE_WAREHOUSE & orig)

Copy constructor

Definition at line 11 of file CARPHONE_WAREHOUSE.cpp.

5.6.2.5 CARPHONE_WAREHOUSE::~~CARPHONE_WAREHOUSE () [virtual]

de-constructor

Definition at line 14 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3 Member Function Documentation

5.6.3.1 void CARPHONE_WAREHOUSE::copy (std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) [virtual]

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 34 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.2 std::shared_ptr< OSTM > CARPHONE_WAREHOUSE::getBaseCopy (std::shared_ptr< OSTM > object) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK* type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a BANK* return type

Reimplemented from [OSTM](#).

Definition at line 21 of file CARPHONE_WAREHOUSE.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.6.3.3 `int CARPHONE_WAREHOUSE::GetNumber_of_alcatel () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 71 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.4 `int CARPHONE_WAREHOUSE::GetNumber_of_huawei () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 87 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.5 `int CARPHONE_WAREHOUSE::GetNumber_of_iphones () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 111 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.6 `int CARPHONE_WAREHOUSE::GetNumber_of_nokia () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 79 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.7 `int CARPHONE_WAREHOUSE::GetNumber_of_samsung () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 103 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.8 `int CARPHONE_WAREHOUSE::GetNumber_of_sony () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 95 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.9 `std::string CARPHONE_WAREHOUSE::GetShop_address () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 127 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.10 `std::string CARPHONE_WAREHOUSE::GetShop_name () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 119 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.11 `CARPHONE_WAREHOUSE CARPHONE_WAREHOUSE::operator= (const CARPHONE_WAREHOUSE & orig) [inline]`

Operator

Definition at line 75 of file CARPHONE_WAREHOUSE.h.

Here is the call graph for this function:

5.6.3.12 `void CARPHONE_WAREHOUSE::SetNumber_of_alcatel (int _number_of_alcatel) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 67 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.13 `void CARPHONE_WAREHOUSE::SetNumber_of_huawei (int _number_of_huawei) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 83 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.14 `void CARPHONE_WAREHOUSE::SetNumber_of_iphones (int _number_of_iphones) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 107 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.15 void CARPHONE_WAREHOUSE::SetNumber_of_nokia (int *_number_of_nokia*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 75 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.16 void CARPHONE_WAREHOUSE::SetNumber_of_samsung (int *_number_of_samsung*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 99 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.17 void CARPHONE_WAREHOUSE::SetNumber_of_sony (int *_number_of_sony*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 91 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.18 void CARPHONE_WAREHOUSE::SetShop_address (std::string *_shop_address*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 123 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.19 void CARPHONE_WAREHOUSE::SetShop_name (std::string *_shop_name*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 115 of file CARPHONE_WAREHOUSE.cpp.

Here is the caller graph for this function:

5.6.3.20 void CARPHONE_WAREHOUSE::toString () [virtual]

`_cast`, is use to cast bak the `std::shared_ptr<OSTM>` to the required type

`toString` function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 60 of file CARPHONE_WAREHOUSE.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [CARPHONE_WAREHOUSE.h](#)
- [CARPHONE_WAREHOUSE.cpp](#)

5.7 DUNDALK_W Class Reference

```
#include <DUNDALK_W.h>
```

Inheritance diagram for DUNDALK_W:

Collaboration diagram for DUNDALK_W:

Public Member Functions

- [DUNDALK_W](#) ()
- [DUNDALK_W](#) (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- [DUNDALK_W](#) (std::shared_ptr< [WAREHOUSE](#) > obj, int _version, int _unique_id)
- [DUNDALK_W](#) (const [DUNDALK_W](#) &orig)
- [DUNDALK_W](#) operator= (const [DUNDALK_W](#) &orig)
- virtual [~DUNDALK_W](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new BANK type object*
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetNumber_of_alcatel](#) (int _number_of_alcatel)
- virtual int [GetNumber_of_alcatel](#) ()
- virtual void [SetNumber_of_nokia](#) (int _number_of_nokia)
- virtual int [GetNumber_of_nokia](#) ()
- virtual void [SetNumber_of_huawei](#) (int _number_of_huawei)
- virtual int [GetNumber_of_huawei](#) ()
- virtual void [SetNumber_of_sony](#) (int _number_of_sony)
- virtual int [GetNumber_of_sony](#) ()
- virtual void [SetNumber_of_samsung](#) (int _number_of_samsung)
- virtual int [GetNumber_of_samsung](#) ()
- virtual void [SetNumber_of_iphones](#) (int _number_of_iphones)
- virtual int [GetNumber_of_iphones](#) ()
- virtual void [SetShop_name](#) (std::string _shop_name)
- virtual std::string [GetShop_name](#) ()
- virtual void [SetShop_address](#) (std::string _shop_address)
- virtual std::string [GetShop_address](#) ()

5.7.1 Detailed Description

Inherit from [WAREHOUSE](#)

Definition at line 19 of file DUNDALK_W.h.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 DUNDALK_W::DUNDALK_W () [inline]

Constructor

Definition at line 24 of file DUNDALK_W.h.

Here is the caller graph for this function:

5.7.2.2 DUNDALK_W::DUNDALK_W (std::string *address*, std::string *shop_name*, int *iphone*, int *samsung*, int *sony*, int *huawei*, int *nokia*, int *alcatel*) [inline]

Custom constructor

Definition at line 38 of file DUNDALK_W.h.

5.7.2.3 DUNDALK_W::DUNDALK_W (std::shared_ptr< WAREHOUSE > *obj*, int *version*, int *unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 55 of file DUNDALK_W.h.

Here is the call graph for this function:

5.7.2.4 DUNDALK_W::DUNDALK_W (const DUNDALK_W & *orig*)

Copy constructor

Definition at line 15 of file DUNDALK_W.cpp.

5.7.2.5 DUNDALK_W::~~DUNDALK_W () [virtual]

de-constructor

Definition at line 12 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3 Member Function Documentation

5.7.3.1 void DUNDALK_W::copy (std::shared_ptr< OSTM > *to*, std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 35 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.2 `std::shared_ptr< OSTM > DUNDALK_W::getBaseCopy (std::shared_ptr< OSTM > object)` [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK* type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a BANK* return type

Reimplemented from [OSTM](#).

Definition at line 22 of file DUNDALK_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.7.3.3 `int DUNDALK_W::GetNumber_of_alcatel ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 73 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.4 `int DUNDALK_W::GetNumber_of_huawei ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 89 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.5 `int DUNDALK_W::GetNumber_of_iphones ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 113 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.6 `int DUNDALK_W::GetNumber_of_nokia () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 81 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.7 `int DUNDALK_W::GetNumber_of_samsung () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 105 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.8 `int DUNDALK_W::GetNumber_of_sony () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 97 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.9 `std::string DUNDALK_W::GetShop_address () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 129 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.10 `std::string DUNDALK_W::GetShop_name () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 121 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.11 `DUNDALK_W DUNDALK_W::operator=(const DUNDALK_W & orig) [inline]`

Operator

Definition at line 75 of file DUNDALK_W.h.

Here is the call graph for this function:

5.7.3.12 `void DUNDALK_W::SetNumber_of_alcatel (int _number_of_alcatel) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 69 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.13 `void DUNDALK_W::SetNumber_of_huawei (int _number_of_huawei) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 85 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.14 `void DUNDALK_W::SetNumber_of_iphones (int _number_of_iphones) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 109 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.15 `void DUNDALK_W::SetNumber_of_nokia (int _number_of_nokia) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 77 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.16 `void DUNDALK_W::SetNumber_of_samsung (int _number_of_samsung) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 101 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.17 `void DUNDALK_W::SetNumber_of_sony (int _number_of_sony) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 93 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.18 `void DUNDALK_W::SetShop_address (std::string_shop_address) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 125 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.19 `void DUNDALK_W::SetShop_name (std::string_shop_name) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 117 of file DUNDALK_W.cpp.

Here is the caller graph for this function:

5.7.3.20 `void DUNDALK_W::toString () [virtual]`

`_cast`, is use to cast bak the `std::shared_ptr<OSTM>` to the required type

`toString` function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 62 of file DUNDALK_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [DUNDALK_W.h](#)
- [DUNDALK_W.cpp](#)

5.8 KILKENNY_W Class Reference

```
#include <KILKENNY_W.h>
```

Inheritance diagram for KILKENNY_W:

Collaboration diagram for KILKENNY_W:

Public Member Functions

- [KILKENNY_W](#) ()
- [KILKENNY_W](#) (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- [KILKENNY_W](#) (std::shared_ptr< [WAREHOUSE](#) > obj, int _version, int _unique_id)
- [KILKENNY_W](#) (const [KILKENNY_W](#) &orig)
- [KILKENNY_W operator=](#) (const [KILKENNY_W](#) &orig)
- virtual [~KILKENNY_W](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new BANK type object*
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetNumber_of_alcatel](#) (int _number_of_alcatel)
- virtual int [GetNumber_of_alcatel](#) ()
- virtual void [SetNumber_of_nokia](#) (int _number_of_nokia)
- virtual int [GetNumber_of_nokia](#) ()
- virtual void [SetNumber_of_huawei](#) (int _number_of_huawei)
- virtual int [GetNumber_of_huawei](#) ()
- virtual void [SetNumber_of_sony](#) (int _number_of_sony)
- virtual int [GetNumber_of_sony](#) ()
- virtual void [SetNumber_of_samsung](#) (int _number_of_samsung)
- virtual int [GetNumber_of_samsung](#) ()
- virtual void [SetNumber_of_iphones](#) (int _number_of_iphones)
- virtual int [GetNumber_of_iphones](#) ()
- virtual void [SetShop_name](#) (std::string _shop_name)
- virtual std::string [GetShop_name](#) ()
- virtual void [SetShop_address](#) (std::string _shop_address)
- virtual std::string [GetShop_address](#) ()

5.8.1 Detailed Description

Inherit from [WAREHOUSE](#)

Definition at line 19 of file [KILKENNY_W.h](#).

5.8.2 Constructor & Destructor Documentation

5.8.2.1 [KILKENNY_W::KILKENNY_W](#) () `[inline]`

Constructor

Definition at line 24 of file [KILKENNY_W.h](#).

Here is the caller graph for this function:

5.8.2.2 `KILKENNY_W::KILKENNY_W (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel) [inline]`

Custom constructor

Definition at line 38 of file KILKENNY_W.h.

5.8.2.3 `KILKENNY_W::KILKENNY_W (std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id) [inline]`

Custom constructor, used by the library for deep copying

Definition at line 55 of file KILKENNY_W.h.

Here is the call graph for this function:

5.8.2.4 `KILKENNY_W::KILKENNY_W (const KILKENNY_W & orig)`

Copy constructor

Definition at line 15 of file KILKENNY_W.cpp.

5.8.2.5 `KILKENNY_W::~~KILKENNY_W () [virtual]`

de-constructor

Definition at line 12 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3 Member Function Documentation

5.8.3.1 `void KILKENNY_W::copy (std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) [virtual]`

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 35 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.2 `std::shared_ptr< OSTM > KILKENNY_W::getBaseCopy (std::shared_ptr< OSTM > object)` [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK* type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a BANK* return type

Reimplemented from [OSTM](#).

Definition at line 22 of file KILKENNY_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.8.3.3 `int KILKENNY_W::GetNumber_of_alcatel ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 73 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.4 `int KILKENNY_W::GetNumber_of_huawei ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 89 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.5 `int KILKENNY_W::GetNumber_of_iphones ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 113 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.6 `int KILKENNY_W::GetNumber_of_nokia ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 81 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.7 `int KILKENNY_W::GetNumber_of_samsung () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 105 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.8 `int KILKENNY_W::GetNumber_of_sony () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 97 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.9 `std::string KILKENNY_W::GetShop_address () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 129 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.10 `std::string KILKENNY_W::GetShop_name () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 121 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.11 `KILKENNY_W KILKENNY_W::operator= (const KILKENNY_W & orig) [inline]`

Operator

Definition at line 75 of file KILKENNY_W.h.

Here is the call graph for this function:

5.8.3.12 `void KILKENNY_W::SetNumber_of_alcatel (int _number_of_alcatel) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 69 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.13 void KILKENNY_W::SetNumber_of_huawei (int *_number_of_huawei*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 85 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.14 void KILKENNY_W::SetNumber_of_iphones (int *_number_of_iphones*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 109 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.15 void KILKENNY_W::SetNumber_of_nokia (int *_number_of_nokia*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 77 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.16 void KILKENNY_W::SetNumber_of_samsung (int *_number_of_samsung*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 101 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.17 void KILKENNY_W::SetNumber_of_sony (int *_number_of_sony*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 93 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.18 void KILKENNY_W::SetShop_address (std::string *_shop_address*) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 125 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.19 void KILKENNY_W::SetShop_name (std::string _shop_name) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 117 of file KILKENNY_W.cpp.

Here is the caller graph for this function:

5.8.3.20 void KILKENNY_W::toString () [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 62 of file KILKENNY_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [KILKENNY_W.h](#)
- [KILKENNY_W.cpp](#)

5.9 OSTM Class Reference

```
#include <OSTM.h>
```

Inheritance diagram for OSTM:

Public Member Functions

- [OSTM](#) ()
OSTM Constructor.
- [OSTM](#) (int _version_number_, int _unique_id_)
OSTM Custom Constructor.
- virtual [~OSTM](#) ()
De-constructor.
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > from, std::shared_ptr< [OSTM](#) > to)
OSTM required virtual method for deep copy.
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
OSTM required virtual method for returning a pointer that is copy of the original pointer.
- virtual void [toString](#) ()
OSTM required virtual method for display object.
- void [Set_Unique_ID](#) (int uniqueID)
setter for unique id
- int [Get_Unique_ID](#) () const

- getter for unique id*
- void [Set_Version](#) (int version)
- setter for version number*
- int [Get_Version](#) () const
- getter for version number*
- void [increase_VersionNumber](#) ()
- commit time increase version number to child object*
- bool [Is_Can_Commit](#) () const
- NOT USED YET.*
- void [Set_Can_Commit](#) (bool canCommit)
- NOT USED YET.*
- void [Set_Abort_Transaction](#) (bool abortTransaction)
- NOT USED YET.*
- bool [Is_Abort_Transaction](#) () const
- NOT USED YET.*
- void [lock_Mutex](#) ()
- object unique lock, locks mutex*
- void [unlock_Mutex](#) ()
- object unique lock, unlocks mutex*
- bool [is_Locked](#) ()
- object unique lock, try locks mutex return boolean value depends on the lock state*

5.9.1 Detailed Description

Definition at line 17 of file OSTM.h.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 OSTM::OSTM ()

[OSTM](#) Constructor.

Default constructor.

Parameters

<i>version</i>	indicates the version number of the inherited child pointer
<i>uniqueID</i>	is a unique identifier assigned to every object registered in OSTM library
<i>canCommit</i>	NOT USED YET
<i>abort_Transaction</i>	NOT USED YET

Definition at line 20 of file OSTM.cpp.

5.9.2.2 OSTM::OSTM (int _version_number_, int _unique_id_)

[OSTM](#) Custom Constructor.

Custom Constructor Used for copy object.

Parameters

<i>version</i>	indicates the version number of the inherited child pointer
<i>uniqueID</i>	is a unique identifier assigned to every object registered in OSTM library
<i>canCommit</i>	NOT USED YET
<i>abort_Transaction</i>	NOT USED YET

Definition at line 36 of file OSTM.cpp.

5.9.2.3 OSTM::~OSTM() [virtual]

De-constructor.

De-constructor

Definition at line 48 of file OSTM.cpp.

5.9.3 Member Function Documentation

5.9.3.1 virtual void OSTM::copy (std::shared_ptr< OSTM > from, std::shared_ptr< OSTM > to) [inline], [virtual]

[OSTM](#) required virtual method for deep copy.

Reimplemented in [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [CARPHONE_WAREHOUSE](#), [DUNDALK_W](#), [KILKEENNY_W](#), [AIB](#), [BOI](#), [BOA](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 34 of file OSTM.h.

5.9.3.2 int OSTM::Get_Unique_ID () const

getter for unique id

Parameters

<i>uniqueID</i>	int
-----------------	-----

Definition at line 73 of file OSTM.cpp.

Here is the caller graph for this function:

5.9.3.3 int OSTM::Get_Version () const

getter for version number

Parameters

<i>version</i>	int
----------------	-----

Definition at line 89 of file OSTM.cpp.

Here is the caller graph for this function:

```
5.9.3.4  virtual std::shared_ptr<OSTM> OSTM::getBaseCopy ( std::shared_ptr< OSTM > object )  [inline],
        [virtual]
```

[OSTM](#) required virtual method for returning a pointer that is copy of the original pointer.

Reimplemented in [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [CARPHONE_WAREHOUSE](#), [DUNDALK_W](#), [KILKEANNY_W](#), [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 38 of file OSTM.h.

```
5.9.3.5  void OSTM::increase_VersionNumber ( )
```

commit time increase version number to child object

Parameters

<i>version</i>	int
----------------	-----

Definition at line 97 of file OSTM.cpp.

Here is the caller graph for this function:

```
5.9.3.6  bool OSTM::Is_Abort_Transaction ( ) const
```

NOT USED YET.

Parameters

<i>abort_Transaction</i>	boolean
--------------------------	---------

Definition at line 126 of file OSTM.cpp.

Here is the caller graph for this function:

```
5.9.3.7  bool OSTM::Is_Can_Commit ( ) const
```

NOT USED YET.

Parameters

<i>canCommit</i>	boolean
------------------	---------

Definition at line 112 of file OSTM.cpp.

Here is the caller graph for this function:

5.9.3.8 bool OSTM::is_Locked ()

object unique lock, try locks mutex return boolean value depends on the lock state

Parameters

<i>mutex</i>	std::mutex
--------------	------------

Definition at line 147 of file OSTM.cpp.

Here is the caller graph for this function:

5.9.3.9 void OSTM::lock_Mutex ()

object unique lock, locks mutex

Parameters

<i>mutex</i>	std::mutex
--------------	------------

Definition at line 133 of file OSTM.cpp.

Here is the caller graph for this function:

5.9.3.10 void OSTM::Set_Abort_Transaction (bool *abortTransaction*)

NOT USED YET.

Parameters

<i>abort_Transaction</i>	boolean
--------------------------	---------

Definition at line 119 of file OSTM.cpp.

Here is the caller graph for this function:

5.9.3.11 void OSTM::Set_Can_Commit (bool *canCommit*)

NOT USED YET.

Parameters

<i>canCommit</i>	boolean
------------------	---------

Definition at line 105 of file OSTM.cpp.

Here is the caller graph for this function:

5.9.3.12 void OSTM::Set_Unique_ID (int *uniqueID*)

setter for unique id

Parameters

<i>uniqueID</i>	int
-----------------	-----

Definition at line 66 of file OSTM.cpp.

Here is the caller graph for this function:

5.9.3.13 void OSTM::Set_Version (int *version*)

setter for version number

Parameters

<i>version</i>	int
----------------	-----

Definition at line 81 of file OSTM.cpp.

Here is the caller graph for this function:

5.9.3.14 virtual void OSTM::toString () [inline],[virtual]

[OSTM](#) required virtual method for display object.

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), [KILKEENNY_W](#), [AIB](#), [BOA](#), [BOI](#), [SWBPLC](#), [ULSTER](#), and [UNBL](#).

Definition at line 42 of file OSTM.h.

Here is the call graph for this function:

5.9.3.15 void OSTM::unlock_Mutex ()

object unique lock, unlocks mutex

Parameters

<code>mutex</code>	<code>std::mutex</code>
--------------------	-------------------------

Definition at line 140 of file OSTM.cpp.

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [OSTM.h](#)
- [OSTM.cpp](#)

5.10 SLIGO_W Class Reference

```
#include <SLIGO_W.h>
```

Inheritance diagram for SLIGO_W:

Collaboration diagram for SLIGO_W:

Public Member Functions

- [SLIGO_W](#) ()
- [SLIGO_W](#) (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- [SLIGO_W](#) (std::shared_ptr< [WAREHOUSE](#) > obj, int _version, int _unique_id)
- [SLIGO_W](#) (const [SLIGO_W](#) &orig)
- [SLIGO_W operator=](#) (const [SLIGO_W](#) &orig)
- virtual [~SLIGO_W](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new BANK type object*
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetNumber_of_alcatel](#) (int _number_of_alcatel)
- virtual int [GetNumber_of_alcatel](#) ()
- virtual void [SetNumber_of_nokia](#) (int _number_of_nokia)
- virtual int [GetNumber_of_nokia](#) ()
- virtual void [SetNumber_of_huawei](#) (int _number_of_huawei)
- virtual int [GetNumber_of_huawei](#) ()
- virtual void [SetNumber_of_sony](#) (int _number_of_sony)
- virtual int [GetNumber_of_sony](#) ()
- virtual void [SetNumber_of_samsung](#) (int _number_of_samsung)
- virtual int [GetNumber_of_samsung](#) ()
- virtual void [SetNumber_of_iphones](#) (int _number_of_iphones)
- virtual int [GetNumber_of_iphones](#) ()
- virtual void [SetShop_name](#) (std::string _shop_name)
- virtual std::string [GetShop_name](#) ()
- virtual void [SetShop_address](#) (std::string _shop_address)
- virtual std::string [GetShop_address](#) ()

5.10.1 Detailed Description

Inherit from [WAREHOUSE](#)

Definition at line 19 of file SLIGO_W.h.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 SLIGO_W::SLIGO_W () [inline]

Constructor

Definition at line 24 of file SLIGO_W.h.

Here is the caller graph for this function:

5.10.2.2 SLIGO_W::SLIGO_W (std::string *address*, std::string *shop_name*, int *iphone*, int *samsung*, int *sony*, int *huawei*, int *nokia*, int *alcatel*) [inline]

Custom constructor

Definition at line 38 of file SLIGO_W.h.

5.10.2.3 SLIGO_W::SLIGO_W (std::shared_ptr< WAREHOUSE > *obj*, int *_version*, int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 55 of file SLIGO_W.h.

Here is the call graph for this function:

5.10.2.4 SLIGO_W::SLIGO_W (const SLIGO_W & *orig*)

Copy constructor

Definition at line 15 of file SLIGO_W.cpp.

5.10.2.5 SLIGO_W::~~SLIGO_W () [virtual]

de-constructor

Definition at line 12 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3 Member Function Documentation

5.10.3.1 void SLIGO_W::copy (std::shared_ptr< OSTM > *to*, std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 35 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.2 `std::shared_ptr< OSTM > SLIGO_W::getBaseCopy (std::shared_ptr< OSTM > object)` [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK* type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a BANK* return type

Reimplemented from [OSTM](#).

Definition at line 22 of file SLIGO_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.10.3.3 `int SLIGO_W::GetNumber_of_alcatel ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 73 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.4 `int SLIGO_W::GetNumber_of_huawei ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 89 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.5 `int SLIGO_W::GetNumber_of_iphones ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 113 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.6 `int SLIGO_W::GetNumber_of_nokia () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 81 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.7 `int SLIGO_W::GetNumber_of_samsung () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 105 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.8 `int SLIGO_W::GetNumber_of_sony () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 97 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.9 `std::string SLIGO_W::GetShop_address () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 129 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.10 `std::string SLIGO_W::GetShop_name () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 121 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.11 `SLIGO_W SLIGO_W::operator= (const SLIGO_W & orig) [inline]`

Operator

Definition at line 75 of file SLIGO_W.h.

Here is the call graph for this function:

5.10.3.12 `void SLIGO_W::SetNumber_of_alcatel (int number_of_alcatel) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 69 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.13 `void SLIGO_W::SetNumber_of_huawei (int number_of_huawei) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 85 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.14 `void SLIGO_W::SetNumber_of_iphones (int number_of_iphones) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 109 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.15 `void SLIGO_W::SetNumber_of_nokia (int number_of_nokia) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 77 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.16 `void SLIGO_W::SetNumber_of_samsung (int number_of_samsung) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 101 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.17 `void SLIGO_W::SetNumber_of_sony (int number_of_sony) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 93 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.18 void SLIGO_W::SetShop_address (std::string_shop_address) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 125 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.19 void SLIGO_W::SetShop_name (std::string_shop_name) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 117 of file SLIGO_W.cpp.

Here is the caller graph for this function:

5.10.3.20 void SLIGO_W::toString () [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 62 of file SLIGO_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [SLIGO_W.h](#)
- [SLIGO_W.cpp](#)

5.11 SWBPLC Class Reference

```
#include <SWBPLC.h>
```

Inheritance diagram for SWBPLC:

Collaboration diagram for SWBPLC:

Public Member Functions

- [SWBPLC](#) ()
- [SWBPLC](#) (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- [SWBPLC](#) (std::shared_ptr< [BANK](#) > obj, int _version, int _unique_id)
- [SWBPLC](#) (const [SWBPLC](#) &orig)
- [SWBPLC operator=](#) (const [SWBPLC](#) &orig)
- virtual [~SWBPLC](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetAddress](#) (std::string address)
- virtual std::string [GetAddress](#) () const
- virtual void [SetBalance](#) (double balance)
- virtual double [GetBalance](#) () const
- virtual void [SetAccountNumber](#) (int accountNumber)
- virtual int [GetAccountNumber](#) () const
- virtual void [SetLastName](#) (std::string lastName)
- virtual std::string [GetLastName](#) () const
- virtual void [SetFirstName](#) (std::string firstName)
- virtual std::string [GetFirstName](#) () const
- virtual void [SetFullname](#) (std::string fullname)
- virtual std::string [GetFullname](#) () const

5.11.1 Detailed Description

Inherit from [BANK](#)

Definition at line 19 of file SWBPLC.h.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 [SWBPLC::SWBPLC](#) () `[inline]`

Constructor

Definition at line 24 of file SWBPLC.h.

Here is the caller graph for this function:

5.11.2.2 [SWBPLC::SWBPLC](#) (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address) `[inline]`

Custom constructor

Definition at line 35 of file SWBPLC.h.

5.11.2.3 SWBPLC::SWBPLC (std::shared_ptr< BANK > obj, int_version, int_unique_id) [inline]

Custom constructor, used by the library for deep copying

Definition at line 46 of file SWBPLC.h.

Here is the call graph for this function:

5.11.2.4 SWBPLC::SWBPLC (const SWBPLC & orig)

Copy constructor

Definition at line 12 of file SWBPLC.cpp.

5.11.2.5 SWBPLC::~~SWBPLC () [virtual]

de-constructor

Definition at line 15 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3 Member Function Documentation

5.11.3.1 void SWBPLC::copy (std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) [virtual]

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 34 of file SWBPLC.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.11.3.2 int SWBPLC::GetAccountNumber () const [virtual]

Reimplemented from [BANK](#).

Definition at line 80 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.3 `std::string SWBPLC::GetAddress () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 64 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.4 `double SWBPLC::GetBalance () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 72 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.5 `std::shared_ptr< OSTM > SWBPLC::getBaseCopy (std::shared_ptr< OSTM > object)` [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new `std::shared_ptr<BANK>` type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a <code>std::shared_ptr<BANK></code> return type

Reimplemented from [OSTM](#).

Definition at line 22 of file SWBPLC.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.11.3.6 `std::string SWBPLC::GetFirstName () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 96 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.7 `std::string SWBPLC::GetFullname () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 104 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.8 `std::string SWBPLC::GetLastName () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 88 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.9 `SWBPLC SWBPLC::operator= (const SWBPLC & orig) [inline]`

Operator

Definition at line 63 of file SWBPLC.h.

Here is the call graph for this function:

5.11.3.10 `void SWBPLC::SetAccountNumber (int accountNumber) [virtual]`

Reimplemented from [BANK](#).

Definition at line 76 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.11 `void SWBPLC::SetAddress (std::string address) [virtual]`

Reimplemented from [BANK](#).

Definition at line 60 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.12 `void SWBPLC::SetBalance (double balance) [virtual]`

Reimplemented from [BANK](#).

Definition at line 68 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.13 `void SWBPLC::SetFirstName (std::string firstName) [virtual]`

Reimplemented from [BANK](#).

Definition at line 92 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.14 `void SWBPLC::SetFullname (std::string fullname) [virtual]`

Reimplemented from [BANK](#).

Definition at line 100 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.15 `void SWBPLC::SetLastName (std::string lastName) [virtual]`

Reimplemented from [BANK](#).

Definition at line 84 of file SWBPLC.cpp.

Here is the caller graph for this function:

5.11.3.16 `void SWBPLC::toString () [virtual]`

`_cast`, is use to cast bak the `std::shared_ptr<OSTM>` to the required type

`toString` function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 55 of file SWBPLC.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [SWBPLC.h](#)
- [SWBPLC.cpp](#)

5.12 TALLAGH_W Class Reference

```
#include <TALLAGH_W.h>
```

Inheritance diagram for TALLAGH_W:

Collaboration diagram for TALLAGH_W:

Public Member Functions

- [TALLAGH_W](#) ()
- [TALLAGH_W](#) (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- [TALLAGH_W](#) (std::shared_ptr< [WAREHOUSE](#) > obj, int _version, int _unique_id)
- [TALLAGH_W](#) (const [TALLAGH_W](#) &orig)
- [TALLAGH_W](#) operator= (const [TALLAGH_W](#) &orig)
- virtual [~TALLAGH_W](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new BANK type object*
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetNumber_of_alcatel](#) (int _number_of_alcatel)
- virtual int [GetNumber_of_alcatel](#) ()
- virtual void [SetNumber_of_nokia](#) (int _number_of_nokia)
- virtual int [GetNumber_of_nokia](#) ()
- virtual void [SetNumber_of_huawei](#) (int _number_of_huawei)
- virtual int [GetNumber_of_huawei](#) ()
- virtual void [SetNumber_of_sony](#) (int _number_of_sony)
- virtual int [GetNumber_of_sony](#) ()
- virtual void [SetNumber_of_samsung](#) (int _number_of_samsung)
- virtual int [GetNumber_of_samsung](#) ()
- virtual void [SetNumber_of_iphones](#) (int _number_of_iphones)
- virtual int [GetNumber_of_iphones](#) ()
- virtual void [SetShop_name](#) (std::string _shop_name)
- virtual std::string [GetShop_name](#) ()
- virtual void [SetShop_address](#) (std::string _shop_address)
- virtual std::string [GetShop_address](#) ()

5.12.1 Detailed Description

Inherit from [WAREHOUSE](#)

Definition at line 19 of file TALLAGH_W.h.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 TALLAGH_W::TALLAGH_W () [inline]

Constructor

Definition at line 24 of file TALLAGH_W.h.

Here is the caller graph for this function:

5.12.2.2 `TALLAGH_W::TALLAGH_W (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel) [inline]`

Custom constructor

Definition at line 38 of file TALLAGH_W.h.

5.12.2.3 `TALLAGH_W::TALLAGH_W (std::shared_ptr< WAREHOUSE > obj, int _version, int _unique_id) [inline]`

Custom constructor, used by the library for deep copying

Definition at line 55 of file TALLAGH_W.h.

Here is the call graph for this function:

5.12.2.4 `TALLAGH_W::TALLAGH_W (const TALLAGH_W & orig)`

Copy constructor

Definition at line 15 of file TALLAGH_W.cpp.

5.12.2.5 `TALLAGH_W::~~TALLAGH_W () [virtual]`

de-constructor

Definition at line 12 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3 Member Function Documentation

5.12.3.1 `void TALLAGH_W::copy (std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) [virtual]`

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a BANK* type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 35 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.2 `std::shared_ptr< OSTM > TALLAGH_W::getBaseCopy (std::shared_ptr< OSTM > object)` [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK* type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a BANK* return type

Reimplemented from [OSTM](#).

Definition at line 22 of file TALLAGH_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.12.3.3 `int TALLAGH_W::GetNumber_of_alcatel ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 71 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.4 `int TALLAGH_W::GetNumber_of_huawei ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 87 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.5 `int TALLAGH_W::GetNumber_of_iphones ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 111 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.6 `int TALLAGH_W::GetNumber_of_nokia ()` [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 79 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.7 `int TALLAGH_W::GetNumber_of_samsung () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 103 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.8 `int TALLAGH_W::GetNumber_of_sony () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 95 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.9 `std::string TALLAGH_W::GetShop_address () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 127 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.10 `std::string TALLAGH_W::GetShop_name () [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 119 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.11 `TALLAGH_W TALLAGH_W::operator= (const TALLAGH_W & orig) [inline]`

Operator

Definition at line 75 of file TALLAGH_W.h.

Here is the call graph for this function:

5.12.3.12 `void TALLAGH_W::SetNumber_of_alcatel (int _number_of_alcatel) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 67 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.13 `void TALLAGH_W::SetNumber_of_huawei (int _number_of_huawei) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 83 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.14 `void TALLAGH_W::SetNumber_of_iphones (int _number_of_iphones) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 107 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.15 `void TALLAGH_W::SetNumber_of_nokia (int _number_of_nokia) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 75 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.16 `void TALLAGH_W::SetNumber_of_samsung (int _number_of_samsung) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 99 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.17 `void TALLAGH_W::SetNumber_of_sony (int _number_of_sony) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 91 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.18 `void TALLAGH_W::SetShop_address (std::string _shop_address) [virtual]`

Reimplemented from [WAREHOUSE](#).

Definition at line 123 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.19 void TALLAGH_W::SetShop_name (std::string_shop_name) [virtual]

Reimplemented from [WAREHOUSE](#).

Definition at line 115 of file TALLAGH_W.cpp.

Here is the caller graph for this function:

5.12.3.20 void TALLAGH_W::toString () [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 62 of file TALLAGH_W.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [TALLAGH_W.h](#)
- [TALLAGH_W.cpp](#)

5.13 TM Class Reference

```
#include <TM.h>
```

Public Member Functions

- std::shared_ptr< [TX](#) > const [_get_tx](#) ()
_get_tx std::shared_ptr<TX>, returning a shared pointer with the transaction
- void [_TX_EXIT](#) ()
_TX_EXIT void, the thread calls the ostm_exit function in the transaction, and clear all elements from the shared global collection associated with the main process
- void [print_all](#) ()
ONLY FOR TESTING print_all void, print out all object key from txMAP collection.

Static Public Member Functions

- static [TM](#) & [Instance](#) ()
Scott Meyer's Singleton creation, what is thread safe.

5.13.1 Detailed Description

Definition at line 58 of file TM.h.

5.13.2 Member Function Documentation

5.13.2.1 std::shared_ptr< [TX](#) > const TM::_get_tx ()

[_get_tx](#) std::shared_ptr<TX>, returning a shared pointer with the transaction

[_get_tx](#) std::shared_ptr<TX>, return a shared_ptr with the Transaction object, if [TX](#) not exists then create one, else increasing the nesting level std::mutex, protect shared collection from critical section

Parameters

<i>guard</i>	std::lock_guard, locks the register_Lock mutex, unlock automatically when goes out of the scope
--------------	-------------------------------------------------------------------------------------------------

Definition at line 78 of file TM.cpp.

Here is the caller graph for this function:

5.13.2.2 void TM::_TX_EXIT ()

_TX_EXIT void, the thread calls the ostm_exit function in the transaction, and clear all elements from the shared global collection associated with the main process

_TX_EXIT void, the thread calls the ostm_exit function in the transaction, and clear all elements from the shared global collection associated with the main process tx [TX](#), local object to function in transaction

Definition at line 101 of file TM.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.13.2.3 TM & TM::Instance () [static]

Scott Meyer's Singleton creation, what is thread safe.

Instance [TM](#), return the same singleton object to any process.

Parameters

<i>_instance</i>	TM , static class reference to the instance of the Transaction Manager class
<i>_instance</i>	ppid, assigning the process id whoever created the Singleton instance

Definition at line 28 of file TM.cpp.

Here is the caller graph for this function:

5.13.2.4 void TM::print_all ()

ONLY FOR TESTING print_all void, print out all object key from txMAP collection.

ONLY FOR TESTING print_all void, prints all object in the txMap

Definition at line 121 of file TM.cpp.

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [TM.h](#)
- [TM.cpp](#)

5.14 TX Class Reference

```
#include <TX.h>
```

Public Member Functions

- [TX](#) (std::thread::id id)
Constructor.
- [~TX](#) ()
De-constructor.
- [TX](#) (const [TX](#) &orig)
Default copy constructor.
- void [ostm_exit](#) ()
Delete all map entries associated with the main process.
- void [_register](#) (std::shared_ptr< [OSTM](#) > object)
Register [OSTM](#) pointer into STM library.
- std::shared_ptr< [OSTM](#) > [load](#) (std::shared_ptr< [OSTM](#) > object)
load std::shared_ptr<OSTM>, returning an std::shared_ptr<OSTM> copy of the original pointer, to work with during transaction life time
- void [store](#) (std::shared_ptr< [OSTM](#) > object)
Store transactional changes.
- bool [commit](#) ()
Commit transactional changes.
- void [_increase_tx_nesting](#) ()
Add [TX](#) nesting level by one.
- void [_decrease_tx_nesting](#) ()
Remove [TX](#) nesting level by one.
- int [getTest_counter](#) ()
getTest_counter TESTING ONLY!!! returning the value of the test_counter stored, number of rollbacks
- void [_print_all_tx](#) ()

Static Public Attributes

- static int [test_counter](#) = 0

Friends

- class [TM](#)

5.14.1 Detailed Description

Definition at line 26 of file TX.h.

5.14.2 Constructor & Destructor Documentation

5.14.2.1 TX::TX (std::thread::id id)

Constructor.

Parameters

<i>transaction_Number</i>	int, to store associated thread
<i>_tx_nesting_level</i>	int, to store and indicate nesting level of transactions within transaction

Definition at line 31 of file TX.cpp.

5.14.2.2 TX::~~TX ()

De-constructor.

Definition at line 38 of file TX.cpp.

5.14.2.3 TX::TX (const TX & orig)

Default copy constructor.

Definition at line 44 of file TX.cpp.

5.14.3 Member Function Documentation

5.14.3.1 void TX::_decrease_tx_nesting ()

Remove TX nesting level by one.

_decrease_tx_nesting decrease the value stored in _tx_nesting_level by one, when outer transactions committing

Parameters

<i>_tx_nesting_level</i>	int
--------------------------	-----

Definition at line 316 of file TX.cpp.

Here is the caller graph for this function:

5.14.3.2 void TX::_increase_tx_nesting ()

Add TX nesting level by one.

_increase_tx_nesting increase the value stored in _tx_nesting_level by one, indicate that the transaction nested

Parameters

<i>_tx_nesting_level</i>	int
--------------------------	-----

Definition at line 307 of file TX.cpp.

5.14.3.3 void TX::_print_all_tx ()

ONLY FOR TESTING CHECK THE MAP AFTER THREAD EXIT AND ALL SHOULD BE DELETED!!!!!!!

Definition at line 346 of file TX.cpp.

5.14.3.4 void TX::_register (std::shared_ptr< OSTM > object)

Register [OSTM](#) pointer into STM library.

register void, receives an std::shared_ptr<OSTM> that point to the original memory space to protect from reca conditions

Parameters

<i>working_Map_collection</i>	std::map, store all the std::shared_ptr<OSTM> pointer in the transaction
<i>main_Process_Map_collection</i>	std::map, store all std::shared_ptr<OSTM> from all transaction, used to lock and compare the objects
<i>process_map_collection</i>	std::map, store all std::shared_ptr<OSTM> unique ID from all transaction, used to delete all pointers used by the main process, from all transaction before the program exit.
<i>std::lock_guard</i>	use register_Lock(mutex) shared lock between all transaction
<i>ppid</i>	int, store main process number

Definition at line 104 of file TX.cpp.

5.14.3.5 bool TX::commit ()

Commit transactional changes.

commit bool, returns boolean value TRUE/FALSE depends on the action taken within the function

Parameters

<i>working_Map_collection</i>	std::map, store all the std::shared_ptr<OSTM> pointer in the transaction
<i>main_Process_Map_collection</i>	std::map, store all std::shared_ptr<OSTM> from all transaction, used to lock and compare the objects
<i>can_Commit</i>	bool, helps to make decision that the transaction can commit or rollback

Definition at line 202 of file TX.cpp.

Here is the call graph for this function:

5.14.3.6 int TX::getTest_counter ()

getTest_counter TESTING ONLY!!! returning the value of the test_counter stored, number of rollbacks

Definition at line 324 of file TX.cpp.

5.14.3.7 `std::shared_ptr<OSTM> TX::load (std::shared_ptr<OSTM> object)`

load `std::shared_ptr<OSTM>`, returning an `std::shared_ptr<OSTM>` copy of the original pointer, to work with during transaction life time

Register [OSTM](#) pointer into STM library

Parameters

<i>working_Map_collection</i>	<code>std::map</code> , store all the <code>std::shared_ptr<OSTM></code> pointer in the transaction
-------------------------------	-----------------------------------------------------------------------------------------------------------

Definition at line 155 of file TX.cpp.

5.14.3.8 `void TX::ostm_exit ()`

Delete all map entries associated with the main process.

`ostm_exit` void, clear all elements from the shared global collections associated with the main process

Parameters

<i>main_Process_Map_collection</i>	<code>std::map</code> , store all <code>std::shared_ptr<OSTM></code> from all transaction shared between multiple processes
<i>process_map_collection</i>	<code>std::map</code> , store all unique id from all transaction within main process DO NOT CALL THIS METHOD EXPLICITLY!!!!!! WILL DELETE ALL PROCESS ASSOCIATED ELEMENTS!!!!

Definition at line 72 of file TX.cpp.

Here is the caller graph for this function:

5.14.3.9 `void TX::store (std::shared_ptr<OSTM> object)`

Store transactional changes.

`store` void, receive an `std::shared_ptr<OSTM>` object to store the changes within the transaction, depends the user action

Parameters

<i>working_Map_collection</i>	<code>std::map</code> , store all the <code>std::shared_ptr<OSTM></code> pointer in the transaction
-------------------------------	-----------------------------------------------------------------------------------------------------------

Definition at line 178 of file TX.cpp.

5.14.4 Friends And Related Function Documentation

5.14.4.1 `friend class TM` [[friend](#)]

Only [TM](#) Transaction Manager can create instance of [TX](#) Transaction

Definition at line 72 of file TX.h.

5.14.5 Member Data Documentation

5.14.5.1 `int TX::test_counter = 0` `[static]`

Parameters

<code>test_counter</code>	int ONLY FOR TESTING!!!
<code>static</code>	Global counter for rollback

Definition at line 80 of file TX.h.

The documentation for this class was generated from the following files:

- [TX.h](#)
- [TX.cpp](#)

5.15 ULSTER Class Reference

```
#include <ULSTER.h>
```

Inheritance diagram for ULSTER:

Collaboration diagram for ULSTER:

Public Member Functions

- [ULSTER](#) ()
- [ULSTER](#) (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- [ULSTER](#) (std::shared_ptr< [BANK](#) > obj, int _version, int _unique_id)
- [ULSTER](#) (const [ULSTER](#) &orig)
- [ULSTER operator=](#) (const [ULSTER](#) &orig)
- virtual [~ULSTER](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetAddress](#) (std::string address)
- virtual std::string [GetAddress](#) () const
- virtual void [SetBalance](#) (double balance)
- virtual double [GetBalance](#) () const
- virtual void [SetAccountNumber](#) (int accountNumber)
- virtual int [GetAccountNumber](#) () const
- virtual void [SetLastName](#) (std::string lastName)
- virtual std::string [GetLastName](#) () const
- virtual void [SetFirstName](#) (std::string firstName)
- virtual std::string [GetFirstName](#) () const
- virtual void [SetFullname](#) (std::string fullname)
- virtual std::string [GetFullname](#) () const

5.15.1 Detailed Description

Inherit from [BANK](#)

Definition at line 19 of file ULSTER.h.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 ULSTER::ULSTER () [inline]

Constructor

Definition at line 24 of file ULSTER.h.

Here is the caller graph for this function:

5.15.2.2 ULSTER::ULSTER (int *accountNumber*, double *balance*, std::string *firstName*, std::string *lastName*, std::string *address*) [inline]

Custom constructor

Definition at line 35 of file ULSTER.h.

5.15.2.3 ULSTER::ULSTER (std::shared_ptr< BANK > *obj*, int *_version*, int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 46 of file ULSTER.h.

Here is the call graph for this function:

5.15.2.4 ULSTER::ULSTER (const ULSTER & *orig*)

Copy constructor

Definition at line 15 of file ULSTER.cpp.

5.15.2.5 ULSTER::~~ULSTER () [virtual]

de-constructor

Definition at line 18 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3 Member Function Documentation

5.15.3.1 void ULSTER::copy (std::shared_ptr< OSTM > *to*, std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a <code>std::shared_ptr<BANK></code> type object casted back from <code>std::shared_ptr<OSTM></code>
<i>objFROM</i>	is a <code>std::shared_ptr<BANK></code> type object casted back from <code>std::shared_ptr<OSTM></code>

Reimplemented from [OSTM](#).

Definition at line 37 of file ULSTER.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.15.3.2 `int ULSTER::GetAccountNumber () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 83 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.3 `std::string ULSTER::GetAddress () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 67 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.4 `double ULSTER::GetBalance () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 75 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.5 `std::shared_ptr< OSTM > ULSTER::getBaseCopy (std::shared_ptr< OSTM > object)` `[virtual]`

getBaseCopy function, make deep copy of the object/pointer and Return a new `std::shared_ptr<BANK>` type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a <code>std::shared_ptr<BANK></code> return type

Reimplemented from [OSTM](#).

Definition at line 25 of file ULSTER.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.15.3.6 `std::string ULSTER::GetFirstName () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 99 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.7 `std::string ULSTER::GetFullname () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 107 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.8 `std::string ULSTER::GetLastName () const` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 91 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.9 `ULSTER ULSTER::operator= (const ULSTER & orig)` `[inline]`

Operator

Definition at line 62 of file ULSTER.h.

Here is the call graph for this function:

5.15.3.10 `void ULSTER::SetAccountNumber (int accountNumber)` `[virtual]`

Reimplemented from [BANK](#).

Definition at line 79 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.11 void ULSTER::SetAddress (std::string *address*) [virtual]

Reimplemented from [BANK](#).

Definition at line 63 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.12 void ULSTER::SetBalance (double *balance*) [virtual]

Reimplemented from [BANK](#).

Definition at line 71 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.13 void ULSTER::SetFirstName (std::string *firstName*) [virtual]

Reimplemented from [BANK](#).

Definition at line 95 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.14 void ULSTER::SetFullname (std::string *fullname*) [virtual]

Reimplemented from [BANK](#).

Definition at line 103 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.15 void ULSTER::SetLastName (std::string *lastName*) [virtual]

Reimplemented from [BANK](#).

Definition at line 87 of file ULSTER.cpp.

Here is the caller graph for this function:

5.15.3.16 void ULSTER::toString () [virtual]

`_cast`, is use to cast bak the `std::shared_ptr<OSTM>` to the required type

`toString` function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 58 of file ULSTER.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [ULSTER.h](#)
- [ULSTER.cpp](#)

5.16 UNBL Class Reference

```
#include <UNBL.h>
```

Inheritance diagram for UNBL:

Collaboration diagram for UNBL:

Public Member Functions

- [UNBL](#) ()
- [UNBL](#) (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- [UNBL](#) (std::shared_ptr< [BANK](#) > obj, int _version, int _unique_id)
- [UNBL](#) (const [UNBL](#) &orig)
- [UNBL operator=](#) (const [UNBL](#) &orig)
- virtual [~UNBL](#) ()
- virtual void [copy](#) (std::shared_ptr< [OSTM](#) > to, std::shared_ptr< [OSTM](#) > from)
copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< [OSTM](#) > [getBaseCopy](#) (std::shared_ptr< [OSTM](#) > object)
getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void [toString](#) ()
_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void [SetAddress](#) (std::string address)
- virtual std::string [GetAddress](#) () const
- virtual void [SetBalance](#) (double balance)
- virtual double [GetBalance](#) () const
- virtual void [SetAccountNumber](#) (int accountNumber)
- virtual int [GetAccountNumber](#) () const
- virtual void [SetLastName](#) (std::string lastName)
- virtual std::string [GetLastName](#) () const
- virtual void [SetFirstName](#) (std::string firstName)
- virtual std::string [GetFirstName](#) () const
- virtual void [SetFullname](#) (std::string fullname)
- virtual std::string [GetFullname](#) () const

5.16.1 Detailed Description

Inherit from [BANK](#)

Definition at line 19 of file UNBL.h.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 UNBL::UNBL () [inline]

Constructor

Definition at line 24 of file UNBL.h.

Here is the caller graph for this function:

5.16.2.2 UNBL::UNBL (int *accountNumber*, double *balance*, std::string *firstName*, std::string *lastName*, std::string *address*)
[inline]

Custom constructor

Definition at line 35 of file UNBL.h.

5.16.2.3 UNBL::UNBL (std::shared_ptr< BANK > *obj*, int *_version*, int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 46 of file UNBL.h.

Here is the call graph for this function:

5.16.2.4 UNBL::UNBL (const UNBL & *orig*)

Copy constructor

Definition at line 11 of file UNBL.cpp.

5.16.2.5 UNBL::~UNBL () [virtual]

de-constructor

Definition at line 14 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3 Member Function Documentation

5.16.3.1 void UNBL::copy (std::shared_ptr< OSTM > *to*, std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

<i>objTO</i>	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>
<i>objFROM</i>	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>

Reimplemented from [OSTM](#).

Definition at line 33 of file UNBL.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.16.3.2 `int UNBL::GetAccountNumber () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 78 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.3 `std::string UNBL::GetAddress () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 62 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.4 `double UNBL::GetBalance () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 70 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.5 `std::shared_ptr< OSTM > UNBL::getBaseCopy (std::shared_ptr< OSTM > object)` [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new `std::shared_ptr<BANK>` type object

Parameters

<i>objTO</i>	is a BANK type pointer for casting
<i>obj</i>	is a <code>std::shared_ptr<BANK></code> return type

Reimplemented from [OSTM](#).

Definition at line 21 of file UNBL.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

5.16.3.6 `std::string UNBL::GetFirstName () const` [virtual]

Reimplemented from [BANK](#).

Definition at line 94 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.7 `std::string UNBL::GetFullName () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 102 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.8 `std::string UNBL::GetLastName () const [virtual]`

Reimplemented from [BANK](#).

Definition at line 86 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.9 `UNBL UNBL::operator=(const UNBL & orig) [inline]`

Operator

Definition at line 62 of file UNBL.h.

Here is the call graph for this function:

5.16.3.10 `void UNBL::SetAccountNumber (int accountNumber) [virtual]`

Reimplemented from [BANK](#).

Definition at line 74 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.11 `void UNBL::SetAddress (std::string address) [virtual]`

Reimplemented from [BANK](#).

Definition at line 58 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.12 `void UNBL::SetBalance (double balance) [virtual]`

Reimplemented from [BANK](#).

Definition at line 66 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.13 void UNBL::SetFirstName (std::string *firstName*) [virtual]

Reimplemented from [BANK](#).

Definition at line 90 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.14 void UNBL::SetFullName (std::string *fullname*) [virtual]

Reimplemented from [BANK](#).

Definition at line 98 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.15 void UNBL::SetLastName (std::string *lastName*) [virtual]

Reimplemented from [BANK](#).

Definition at line 82 of file UNBL.cpp.

Here is the caller graph for this function:

5.16.3.16 void UNBL::toString () [virtual]

`_cast`, is use to cast bak the `std::shared_ptr<OSTM>` to the required type

`toString` function, displays the object values in formatted way

Reimplemented from [OSTM](#).

Definition at line 53 of file UNBL.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- [UNBL.h](#)
- [UNBL.cpp](#)

5.17 WAREHOUSE Class Reference

```
#include <WAREHOUSE.h>
```

Inheritance diagram for WAREHOUSE:

Collaboration diagram for WAREHOUSE:

Public Member Functions

- [WAREHOUSE](#) ()
- [WAREHOUSE](#) (int _version, int _unique_id)
- [WAREHOUSE](#) (const [WAREHOUSE](#) &orig)
- virtual [~WAREHOUSE](#) ()
- virtual void [SetNumber_of_alcatel](#) (int _number_of_alcatel)
- virtual int [GetNumber_of_alcatel](#) ()
- virtual void [SetNumber_of_nokia](#) (int _number_of_nokia)
- virtual int [GetNumber_of_nokia](#) ()
- virtual void [SetNumber_of_huawei](#) (int _number_of_huawei)
- virtual int [GetNumber_of_huawei](#) ()
- virtual void [SetNumber_of_sony](#) (int _number_of_sony)
- virtual int [GetNumber_of_sony](#) ()
- virtual void [SetNumber_of_samsung](#) (int _number_of_samsung)
- virtual int [GetNumber_of_samsung](#) ()
- virtual void [SetNumber_of_iphones](#) (int _number_of_iphones)
- virtual int [GetNumber_of_iphones](#) ()
- virtual void [SetShop_name](#) (std::string _shop_name)
- virtual std::string [GetShop_name](#) ()
- virtual void [SetShop_address](#) (std::string _shop_address)
- virtual std::string [GetShop_address](#) ()

5.17.1 Detailed Description

[WAREHOUSE](#) inherit from [OSTM](#) library

Definition at line 16 of file [WAREHOUSE.h](#).

5.17.2 Constructor & Destructor Documentation

5.17.2.1 [WAREHOUSE::WAREHOUSE \(\)](#) `[inline]`

Constructor

Definition at line 21 of file [WAREHOUSE.h](#).

Here is the caller graph for this function:

5.17.2.2 [WAREHOUSE::WAREHOUSE \(int _version, int _unique_id \)](#) `[inline]`

Custom Constructor

Definition at line 27 of file [WAREHOUSE.h](#).

Here is the call graph for this function:

5.17.2.3 WAREHOUSE::WAREHOUSE (const WAREHOUSE & orig)

Copy constructor

Definition at line 12 of file WAREHOUSE.cpp.

5.17.2.4 WAREHOUSE::~~WAREHOUSE () [virtual]

de-constructor

Definition at line 15 of file WAREHOUSE.cpp.

Here is the caller graph for this function:

5.17.3 Member Function Documentation

5.17.3.1 virtual int WAREHOUSE::GetNumber_of_alcatel () [inline],[virtual]

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 44 of file WAREHOUSE.h.

5.17.3.2 virtual int WAREHOUSE::GetNumber_of_huawei () [inline],[virtual]

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 48 of file WAREHOUSE.h.

5.17.3.3 virtual int WAREHOUSE::GetNumber_of_iphones () [inline],[virtual]

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 54 of file WAREHOUSE.h.

5.17.3.4 virtual int WAREHOUSE::GetNumber_of_nokia () [inline],[virtual]

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 46 of file WAREHOUSE.h.

5.17.3.5 `virtual int WAREHOUSE::GetNumber_of_samsung () [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 52 of file WAREHOUSE.h.

5.17.3.6 `virtual int WAREHOUSE::GetNumber_of_sony () [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 50 of file WAREHOUSE.h.

5.17.3.7 `virtual std::string WAREHOUSE::GetShop_address () [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 58 of file WAREHOUSE.h.

5.17.3.8 `virtual std::string WAREHOUSE::GetShop_name () [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 56 of file WAREHOUSE.h.

5.17.3.9 `virtual void WAREHOUSE::SetNumber_of_alcatel (int _number_of_alcatel) [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 43 of file WAREHOUSE.h.

5.17.3.10 `virtual void WAREHOUSE::SetNumber_of_huawei (int _number_of_huawei) [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 47 of file WAREHOUSE.h.

5.17.3.11 `virtual void WAREHOUSE::SetNumber_of_iphones (int _number_of_iphones) [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 53 of file WAREHOUSE.h.

5.17.3.12 `virtual void WAREHOUSE::SetNumber_of_nokia (int _number_of_nokia) [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 45 of file WAREHOUSE.h.

Here is the caller graph for this function:

5.17.3.13 `virtual void WAREHOUSE::SetNumber_of_samsung (int _number_of_samsung) [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 51 of file WAREHOUSE.h.

5.17.3.14 `virtual void WAREHOUSE::SetNumber_of_sony (int _number_of_sony) [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 49 of file WAREHOUSE.h.

5.17.3.15 `virtual void WAREHOUSE::SetShop_address (std::string _shop_address) [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 57 of file WAREHOUSE.h.

5.17.3.16 `virtual void WAREHOUSE::SetShop_name (std::string _shop_name) [inline], [virtual]`

Reimplemented in [CARPHONE_WAREHOUSE](#), [SLIGO_W](#), [TALLAGH_W](#), [CARLOW_W](#), [DUNDALK_W](#), and [KILKENNY_W](#).

Definition at line 55 of file WAREHOUSE.h.

The documentation for this class was generated from the following files:

- [WAREHOUSE.h](#)
- [WAREHOUSE.cpp](#)

Chapter 6

File Documentation

6.1 AIB.cpp File Reference

```
#include <math.h>
#include "AIB.h"
Include dependency graph for AIB.cpp:
```

6.2 AIB.h File Reference

```
#include "BANK.h"
#include <string>
#include <memory>
#include <iostream>
Include dependency graph for AIB.h: This graph shows which files directly or indirectly include this file:
```

Classes

- class [AIB](#)

6.3 BANK.cpp File Reference

```
#include "BANK.h"
Include dependency graph for BANK.cpp:
```

6.4 BANK.h File Reference

```
#include "OSTM.h"
Include dependency graph for BANK.h: This graph shows which files directly or indirectly include this file:
```

Classes

- class [BANK](#)

6.5 BOA.cpp File Reference

```
#include "BOA.h"
```

Include dependency graph for BOA.cpp:

6.6 BOA.h File Reference

```
#include "BANK.h"
#include <string>
#include <memory>
#include <iostream>
```

Include dependency graph for BOA.h: This graph shows which files directly or indirectly include this file:

Classes

- class [BOA](#)

6.7 BOI.cpp File Reference

```
#include "BOI.h"
```

Include dependency graph for BOI.cpp:

6.8 BOI.h File Reference

```
#include "BANK.h"
#include <string>
#include <memory>
#include <iostream>
```

Include dependency graph for BOI.h: This graph shows which files directly or indirectly include this file:

Classes

- class [BOI](#)

6.9 CARLOW_W.cpp File Reference

```
#include "CARLOW_W.h"
```

Include dependency graph for CARLOW_W.cpp:

6.10 CARLOW_W.h File Reference

```
#include "WAREHOUSE.h"  
#include <string>  
#include <memory>  
#include <iostream>
```

Include dependency graph for CARLOW_W.h: This graph shows which files directly or indirectly include this file:

Classes

- class [CARLOW_W](#)

6.11 CARPHONE_WAREHOUSE.cpp File Reference

```
#include "CARPHONE_WAREHOUSE.h"
```

Include dependency graph for CARPHONE_WAREHOUSE.cpp:

6.12 CARPHONE_WAREHOUSE.h File Reference

```
#include "WAREHOUSE.h"  
#include <string>  
#include <memory>  
#include <iostream>
```

Include dependency graph for CARPHONE_WAREHOUSE.h: This graph shows which files directly or indirectly include this file:

Classes

- class [CARPHONE_WAREHOUSE](#)

6.13 DUNDALK_W.cpp File Reference

```
#include "DUNDALK_W.h"
```

Include dependency graph for DUNDALK_W.cpp:

6.14 DUNDALK_W.h File Reference

```
#include "WAREHOUSE.h"  
#include <string>  
#include <memory>  
#include <iostream>
```

Include dependency graph for DUNDALK_W.h: This graph shows which files directly or indirectly include this file:

Classes

- class [DUNDALK_W](#)

6.15 KILKENNY_W.cpp File Reference

```
#include "KILKENNY_W.h"
```

Include dependency graph for KILKENNY_W.cpp:

6.16 KILKENNY_W.h File Reference

```
#include "WAREHOUSE.h"
```

```
#include <string>
```

```
#include <memory>
```

```
#include <iostream>
```

Include dependency graph for KILKENNY_W.h: This graph shows which files directly or indirectly include this file:

Classes

- class [KILKENNY_W](#)

6.17 main.cpp File Reference

```
#include <cstdlib>
```

```
#include <iostream>
```

```
#include <thread>
```

```
#include "TM.h"
```

```
#include "AIB.h"
```

```
#include "BOI.h"
```

```
#include "BOA.h"
```

```
#include "SWBPLC.h"
```

```
#include "ULSTER.h"
```

```
#include "UNBL.h"
```

```
#include "WAREHOUSE.h"
```

```
#include "CARPHONE_WAREHOUSE.h"
```

```
#include "CARLOW_W.h"
```

```
#include "KILKENNY_W.h"
```

```
#include "TALLAGH_W.h"
```

```
#include "DUNDALK_W.h"
```

```
#include "SLIGO_W.h"
```

```
#include <mutex>
```

```
#include <memory>
```

```
#include <condition_variable>
```

```
#include <vector>
```

Include dependency graph for main.cpp:

Functions

- void `_six_account_transfer_` (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_one_, std::shared_ptr< OSTM > _from_two_, std::shared_ptr< OSTM > _from_three_, std::shared_ptr< OSTM > _from_four_, std::shared_ptr< OSTM > _from_five_, TM & tm, double _amount)
six_account_transfer function, takes six std::shared_ptr<OSTM> pointer, the Transaction manager, and the amount to use in the transaction and transfer the _amount value from five account to one account
- void `_two_account_transfer_` (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM & tm, double _amount)
two_account_transfer function, takes two std::shared_ptr<OSTM> pointer, the Transaction manager, and the amount to use in the transaction and transfer the _amount value from one account to the another account
- void `_nesting_` (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM & tm, double _amount)
nesting function, takes two std::shared_ptr<OSTM> pointer, the Transaction manager, and the amount to use in the transaction and transfer the _amount value from one account to the another account This function create nested transactions inside the transaction, and call other function to nesting the transaction as well
- void `_complex_transfer_` (std::shared_ptr< OSTM > _from_, std::shared_ptr< OSTM > _from_two_, std::vector< std::shared_ptr< OSTM >> _customer_vec, TM & tm, double _amount)
complex_transfer function, takes two std::shared_ptr<OSTM> pointer, a vector of std::shared_ptr<OSTM> pointers, the Transaction manager, and the amount to use in the transaction, and transfer the _amount value from booth single objects to the objects to the vector collection
- void `_warehouse_transfer_` (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM & tm, double _amount)
warehouse_transfer function, takes two std::shared_ptr<OSTM> pointer, the Transaction manager, and the amount to use in the transaction and transfer the _amount value from one account to the another account
- void `_nested_warehouse_transfer_` (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _to_two_, std::shared_ptr< OSTM > _to_three_, std::shared_ptr< OSTM > _from_, TM & tm, double _amount)
nested_warehouse_transfer function, takes three std::shared_ptr<OSTM> pointer, the Transaction manager, and the amount to use in the transaction and transfer the _amount value from one account to the another account
- void `_complex_warehouse_transfer_` (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _to_two_, std::shared_ptr< OSTM > _to_three_, std::vector< std::shared_ptr< OSTM >> _warehouse_vec, std::shared_ptr< OSTM > _from_, TM & tm, double _amount)
- int `main` (void)

6.17.1 Function Documentation

6.17.1.1 void `_complex_transfer_` (std::shared_ptr< OSTM > _from_, std::shared_ptr< OSTM > _from_two_, std::vector< std::shared_ptr< OSTM >> _customer_vec, TM & tm, double _amount)

complex_transfer function, takes two std::shared_ptr<OSTM> pointer, a vector of std::shared_ptr<OSTM> pointers, the Transaction manager, and the amount to use in the transaction, and transfer the _amount value from booth single objects to the objects to the vector collection

Parameters

<code>std::shared_ptr< TX ></code>	tx, Transaction Object
<code>std::shared_ptr< BANK ></code>	type, <i>FROM</i> & <i>FROM_TWO</i> & <i>TO</i>
<code>std::shared_ptr< OSTM ></code>	type, <i>FROM_OSTM_ONE</i> & <i>FROM_OSTM_TWO</i> & <i>TO_OSTM</i>

Register the two single account

Declare required pointers

Register customers accounts from the collection (vector)

From `std::shared_ptr<OSTM>` to `std::shared_ptr<BANK>` to access the virtual methods

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

Commit changes

Definition at line 294 of file main.cpp.

Here is the call graph for this function:

```
6.17.1.2 void _complex_warehouse_transfer_ ( std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM >
      _to_two, std::shared_ptr< OSTM > _to_three, std::vector< std::shared_ptr< OSTM >> _warehouse_vec,
      std::shared_ptr< OSTM > _from_, TM & _tm, double _amount )
```

Register the two single account

Declare required pointers

Register customers accounts from the collection (vector)

From `std::shared_ptr<OSTM>` to `std::shared_ptr<BANK>` to access the virtual methods

Make changes with the objects

From `std::shared_ptr<WAREHOUSE>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

NESTED [WAREHOUSE](#) TEST `_to_two`

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

Commit changes

Definition at line 518 of file main.cpp.

Here is the call graph for this function:

```
6.17.1.3 void _nested_warehouse_transfer_ ( std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _to_two,
      std::shared_ptr< OSTM > _to_three, std::shared_ptr< OSTM > _from_, TM & _tm, double _amount )
```

nested_warehouse_transfer function, takes three `std::shared_ptr<OSTM>` pointer, the Transaction manager, and the amount to use in the transaction and transfer the `_amount` value from one account to the another account

Parameters

<code>std::shared_ptr<TX></code>	tx, Transaction Object
<code>std::shared_ptr<WAREHOUSE></code>	type, <i>TO_SHOP</i> & <i>FROM_DIST</i>
<code>std::shared_ptr<OSTM></code>	type, <i>TO_OSTM</i> & <i>FROM_OSTM</i>

Register the two single account

Declare required pointers

From `std::shared_ptr<OSTM>` to `std::shared_ptr<BANK>` to access the virtual methods

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

NESTED [WAREHOUSE](#) TEST `_to_two`

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

Commit changes

Definition at line 419 of file main.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

6.17.1.4 `void _nesting_ (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM & _tm, double _amount)`

nesting function, takes two `std::shared_ptr<OSTM>` pointer, the Transaction manager, and the amount to use in the transaction and transfer the `_amount` value from one account to the another account This function create nested transactions inside the transaction, and call other function to nesting the transaction as well

Parameters

<code>std::shared_ptr<TX></code>	tx, Transaction Object
<code>std::shared_ptr<BANK></code>	type, <i>TO_BANK</i> & <i>FROM_BANK</i>
<code>std::shared_ptr<OSTM></code>	type, <i>TO_OSTM</i> & <i>FROM_OSTM</i>

Register the two single account

Declare required pointers

From `std::shared_ptr<OSTM>` to `std::shared_ptr<BANK>` to access the virtual methods

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

NESTED TRANSACTION

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

NESTED TRANSACTION IN THE NESTED TRANSACTION *two_account_transfer* function call

Commit changes

Definition at line 206 of file main.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

6.17.1.5 `void _six_account_transfer_ (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_one_,
std::shared_ptr< OSTM > _from_two_, std::shared_ptr< OSTM > _from_three_, std::shared_ptr< OSTM >
_from_four_, std::shared_ptr< OSTM > _from_five_, TM & _tm, double _amount)`

six_account_transfer function, takes six `std::shared_ptr<OSTM>` pointer, the Transaction manager, and the amount to use in the transaction and transfer the `_amount` value from five account to one account

Parameters

<code>std::shared_ptr< TX ></code>	tx, Transaction Object
<code>std::shared_ptr< BANK ></code>	type, <i>TO</i> & <i>FROM_ONE</i> & <i>FROM_TWO</i> & <i>FROM_THREE</i> & <i>FROM_FOUR</i> & <i>FROM_FIVE</i>
<code>std::shared_ptr< OSTM ></code>	type, <i>_TO_OSTM</i> & <i>_FROM_ONE_OSTM</i> & <i>_FROM_TWO_OSTM</i> & <i>_FROM_THREE_OSTM</i> & <i>_FROM_FOUR_OSTM</i> & <i>_FROM_FIVE_OSTM</i>

Register the two single account

Required pointers to use in transaction

From `std::shared_ptr<OSTM>` to `std::shared_ptr<BANK>` to access the virtual methods

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

Commit changes

Definition at line 51 of file main.cpp.

Here is the call graph for this function:

6.17.1.6 `void _two_account_transfer_ (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM & _tm, double _amount)`

two_account_transfer function, takes two `std::shared_ptr<OSTM>` pointer, the Transaction manager, and the amount to use in the transaction and transfer the `_amount` value from one account to the another account

Parameters

<code>std::shared_ptr< TX></code>	tx, Transaction Object
<code>std::shared_ptr< BANK></code>	type, <i>TO_BANK</i> & <i>FROM_BANK</i>
<code>std::shared_ptr< OSTM></code>	type, <i>TO_OSTM</i> & <i>FROM_OSTM</i>

Register the two single account

Declare required pointers

From `std::shared_ptr<OSTM>` to `std::shared_ptr<BANK>` to access the virtual methods

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

NESTED TRANSACTION

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

Commit changes

Commit changes

Definition at line 123 of file main.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

6.17.1.7 `void _warehouse_transfer_ (std::shared_ptr< OSTM > _to_, std::shared_ptr< OSTM > _from_, TM & _tm, double _amount)`

warehouse_transfer function, takes two `std::shared_ptr<OSTM>` pointer, the Transaction manager, and the amount to use in the transaction and transfer the `_amount` value from one account to the another account

Parameters

<code>std::shared_ptr< TX></code>	tx, Transaction Object
<code>std::shared_ptr< WAREHOUSE></code>	type, <i>TO_SHOP</i> & <i>FROM_DIST</i>
<code>std::shared_ptr< OSTM></code>	type, <i>TO_OSTM</i> & <i>FROM_OSTM</i>

Register the two single account

Declare required pointers

From `std::shared_ptr<OSTM>` to `std::shared_ptr<BANK>` to access the virtual methods

Make changes with the objects

From `std::shared_ptr<BANK>` to `std::shared_ptr<OSTM>` to store the memory spaces

Store changes

Commit changes

Definition at line 358 of file main.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

6.17.1.8 int main (void)

main method to run test Get the Transaction Manager

```
TM& tm = TM::Instance();
```

Create vector to store `std::shared_ptr<OSTM>` pointers. All object will have unique ID by default

```
std::vector<std::shared_ptr<OSTM>> _customer_vec(vector_number);
std::vector<std::shared_ptr<OSTM>> _warehouse_vec(vector_number);
```

Create objects type of `BANK`. All object will have unique ID by default

```
std::shared_ptr<OSTM> aib_ptr = new AIB(100, 500, "Joe", "Blog", "High street, Kilkenny, Co.Kilkenny");
std::shared_ptr<OSTM> boi_ptr = new BOI(200, 500, "Joe", "Blog", "High street, Kilkenny, Co.Kilkenny");
std::shared_ptr<OSTM> boa_ptr = new BOA(300, 500, "Joe", "Blog", "High street, Kilkenny, Co.Kilkenny");
std::shared_ptr<OSTM> swplc_ptr = new SWBPLC(400, 500, "Joe", "Blog", "High street, Kilkenny, Co.Kilkenny");
std::shared_ptr<OSTM> ulster_ptr = new ULSTER(500, 500, "Joe", "Blog", "High street, Kilkenny, Co.Kilkenny");
std::shared_ptr<OSTM> unbl_ptr = new UNBL(600, 500, "Joe", "Blog", "High street, Kilkenny, Co.Kilkenny");
```

Create objects type of `WAREHOUSE`. All object will have unique ID by default

```
std::shared_ptr<OSTM> w_dist = new CARPHONE_WAREHOUSE();
std::shared_ptr<OSTM> c_shop = new CARLOW_W();
std::shared_ptr<OSTM> k_shop = new KILKENNY_W();
std::shared_ptr<OSTM> t_shop = new TALLAGH_W();
std::shared_ptr<OSTM> d_shop = new DUNDALK_W();
std::shared_ptr<OSTM> s_shop = new SLIGO_W();
```

Create vector of `std::shared_ptr<OSTM>` `BANK` pointers

vector_number is 100 at the moment

```
for(int i=0;i<vector_number;++i)
```

Create vector of `std::shared_ptr<OSTM>` `WAREHOUSE` pointers

vector_number is 100 at the moment

```
for(int i=0;i<vector_number;++i)
```


Display **WAREHOUSE** objects before transaction

```
w_dist->toString();
c_shop->toString();
k_shop->toString();
t_shop->toString();
d_shop->toString();
s_shop->toString();
```

Display **BANK** objects before transaction

```
aib_ptr->toString();
boi_ptr->toString();
boa_ptr->toString();
swplc_ptr->toString();
ulster_ptr->toString();
unbl_ptr->toString();
```

Parameters

<i>transferAmount</i>	in the transaction, control the value in the transaction between objetcs
<i>threadArraySize</i>	control number of threads The logic in the IF ELSE statement distribute the threads between three different thread creating option. If the threadArraySize is divisible with three, the threads will be distributed between function. However, you can creates any number of threads, but to follow the correct output should increase the IF ELSE statement to distribute the threads in equal number.

Creating threadsⁿ -> threadArraySize
for (int i = 0; i < threadArraySize; ++i)

TEST 1 : Nested transaction Test

```
thArray[i] = std::thread(nesting, aib_ptr, boi_ptr, std::ref(tm), transferAmount);
```

TEST 2 :Three different type of function call where the objects are participating in multiple type of transactions

```
thArray[i] = std::thread(two_account_transfer, aib_ptr, boi_ptr, std::ref(tm), transferAmount);
thArray[i] = std::thread(six_account_transfer, boi_ptr, boa_ptr, swplc_ptr, ulster_ptr, aib_ptr, unbl_ptr, std::ref(tm), transferAmount);
thArray[i] = std::thread(complex_transfer, aib_ptr, boi_ptr, std::ref(_customer_vec), std::ref(tm), transferAmount);
```

TEST 3 : Testing **WAREHOUSE** type pointers within transactions

```
thArray[i] = std::thread(phone_transfer, c_shop, w_dist, std::ref(tm), transferAmount);
```

TEST 4 : Testing **WAREHOUSE** type pointers within nested transactions

```
thArray[i] = std::thread(nested_warehouse_transfer, c_shop, d_shop, k_shop, w_dist, std::ref(tm), transferAmount);
```

TEST 5 : Testing **WAREHOUSE** type pointers within mixed and nested transactions

```
thArray[i] = std::thread(warehouse_transfer, c_shop, w_dist, std::ref(tm), transferAmount);
thArray[i] = std::thread(nested_warehouse_transfer, c_shop, d_shop, k_shop, w_dist, std::ref(tm), transferAmount);
thArray[i] = std::thread(complex_warehouse_transfer, d_shop, c_shop, std::ref(_warehouse_vec), w_dist, std::ref(tm), transferAmount);
```

Display objects after all transactions are finished

Uncomment the required corresponding TEST to display results

Extra tx to call and display ROLLBACK value
`std::shared_ptr<TX> tx = tm._get_tx();`

Display the number of ROLLBACK by all the threads
`std::cout << "Rollback counter is : " << tx->getTest_counter() << std::endl;`

Display object from vector

Clean up Transaction Manager from all main process associated transactions
`tm._TX_EXIT();`

Display all Transactions associated with the main process. It should be empty after `_TX_EXIT()` function call!!!
`tm.print_all();`

Definition at line 649 of file main.cpp.

Here is the call graph for this function:

6.18 OSTM.cpp File Reference

```
#include "OSTM.h"
```

Include dependency graph for OSTM.cpp:

6.19 OSTM.h File Reference

```
#include <mutex>
#include <memory>
#include <string>
#include <iostream>
```

Include dependency graph for OSTM.h: This graph shows which files directly or indirectly include this file:

Classes

- class [OSTM](#)

6.20 SLIGO_W.cpp File Reference

```
#include "SLIGO_W.h"
```

Include dependency graph for SLIGO_W.cpp:

6.21 SLIGO_W.h File Reference

```
#include "WAREHOUSE.h"
#include <string>
#include <memory>
#include <iostream>
```

Include dependency graph for SLIGO_W.h: This graph shows which files directly or indirectly include this file:

Classes

- class [SLIGO_W](#)

6.22 SWBPLC.cpp File Reference

```
#include "SWBPLC.h"
```

Include dependency graph for SWBPLC.cpp:

6.23 SWBPLC.h File Reference

```
#include "BANK.h"
```

```
#include <string>
```

```
#include <memory>
```

```
#include <iostream>
```

Include dependency graph for SWBPLC.h: This graph shows which files directly or indirectly include this file:

Classes

- class [SWBPLC](#)

6.24 TALLAGH_W.cpp File Reference

```
#include "TALLAGH_W.h"
```

Include dependency graph for TALLAGH_W.cpp:

6.25 TALLAGH_W.h File Reference

```
#include "WAREHOUSE.h"
```

```
#include <string>
```

```
#include <memory>
```

```
#include <iostream>
```

Include dependency graph for TALLAGH_W.h: This graph shows which files directly or indirectly include this file:

Classes

- class [TALLAGH_W](#)

6.26 TM.cpp File Reference

```
#include "TM.h"
```

```
#include <thread>
```

```
#include <unistd.h>
```

```
#include <sys/types.h>
```

```
#include <iostream>
```

Include dependency graph for TM.cpp:

6.27 TM.h File Reference

```
#include <thread>
#include <mutex>
#include <unordered_map>
#include <utility>
#include <map>
#include "TX.h"
```

Include dependency graph for TM.h: This graph shows which files directly or indirectly include this file:

Classes

- class [TM](#)

6.28 TX.cpp File Reference

```
#include "TX.h"
#include <iostream>
```

Include dependency graph for TX.cpp:

6.29 TX.h File Reference

```
#include <cstdlib>
#include <utility>
#include <map>
#include <iostream>
#include <mutex>
#include <unistd.h>
#include <memory>
#include <stdio.h>
#include <thread>
#include "OSTM.h"
```

Include dependency graph for TX.h: This graph shows which files directly or indirectly include this file:

Classes

- class [TX](#)

6.30 ULSTER.cpp File Reference

```
#include "ULSTER.h"
```

Include dependency graph for ULSTER.cpp:

6.31 ULSTER.h File Reference

```
#include "BANK.h"
#include <string>
#include <memory>
#include <iostream>
```

Include dependency graph for ULSTER.h: This graph shows which files directly or indirectly include this file:

Classes

- class [ULSTER](#)

6.32 UNBL.cpp File Reference

```
#include "UNBL.h"
```

Include dependency graph for UNBL.cpp:

6.33 UNBL.h File Reference

```
#include "BANK.h"
#include <string>
#include <memory>
#include <iostream>
```

Include dependency graph for UNBL.h: This graph shows which files directly or indirectly include this file:

Classes

- class [UNBL](#)

6.34 WAREHOUSE.cpp File Reference

```
#include "WAREHOUSE.h"
```

Include dependency graph for WAREHOUSE.cpp:

6.35 WAREHOUSE.h File Reference

```
#include "OSTM.h"
```

Include dependency graph for WAREHOUSE.h: This graph shows which files directly or indirectly include this file:

Classes

- class [WAREHOUSE](#)

