



Installation and Configuration Manual

Advanced Applications Module (workflow configuration)

03.50.30

29.04.2021

Contents

1	Advanced Applications Module Purpose and Operating Principle	4
2	Service Catalogues	6
2.1	Catalogue of Objects	6
2.2	Catalogue of Actions with Objects	7
2.3	Application Additional Information Classifiers (Application Info Types)	9
2.4	Product Codes	10
3	Advanced Applications Module Configurations	12
3.1	Service Settings	12
3.2	User Administration	13
3.3	Domains	14
3.4	Application Parameters	15
3.5	Workflow Activities	18
3.6	Workflow Strategies	23
3.7	Selecting a Strategy	26
4	Configuring Application Schemes	29
4.1	Application Section Types	29
4.2	Application Schemes	31
4.3	Rules for Building Schemes	35
4.4	Examples of Creating Application Schemes with a Hierarchy of Contracts for Different Clients	38

The Way4™ Advanced Applications R2 module is used to create new system objects (clients, contracts, etc.) and modify parameters of existing objects. It does so by processing manually entered applications, as well as those imported from files or online.

The document describes settings required for operation of the Advanced Applications R2 module.

The following notation can be used in the document:

- Field labels in screen forms are shown in *italics*.
- Key combinations are shown in angular brackets, for example, <Ctrl>+<F3>.
- Names of screen form buttons and tabs are shown in square brackets, for example, [Approve].
- Sequences for selecting user menu items or context menu items are shown using arrows as follows: "Issuing → Contracts Input & Update".
- Sequences for selecting system menu items are shown using arrows as follows: Database => Change password.
- Variables that differ for each local instance, such as directory and file names, as well as file paths are shown in angular brackets, as in <OWS_HOME>.

Warnings and information are marked as follows:



Warnings about potentially hazardous situations or actions.



Messages with information about important features, additional options, or the best use of certain system functions.

1 Advanced Applications Module Purpose and Operating Principle

Applications are processed according to a selected workflow sequence (strategy).

A strategy is an earlier configured sequence of workflow stages.

Within the strategy, earlier configured workflow activities are registered, as well as rules. Rules are a set of conditions and workflow activities when these conditions are met.

Strategy components are workflow stages (steps) that can also be characterized by a set of activities and rules used within the step.

Within a strategy, steps are registered in a specific sequence, corresponding to the workflow algorithm for a particular application. For example, the sequence of operations to process an application for issuing a client debit bankcard includes entry of data to register the client and card contract, verifying the entered data and the operation to issue the card itself. When processing an application to issue a credit bankcard, a client scoring procedure is a required workflow step. A workflow algorithm is determined by bank regulations.

For the strategy as a whole, as well as for separate steps, rules are configured ensuring that when the set conditions are met, the application is sent to another step of the strategy or to the first step of another strategy.

After registering an application as a result of manual input or import from a file, the system selects the workflow strategy. Selection is made according to the application parameter values.

Applications are processed without direct user access to database system tables (clients, contracts and documents). During application processing, all changes to application parameters are reflected only in module tables.

When an application has been processed, a special Accept procedure transfers data from the Advanced Applications module directly into tables of other system modules for registration of contracts, clients, etc.

When an application has been processed, it is given the "Posted" status, after which the application is filtered from the list of applications to be processed.

Applications created using screen forms are built according to a template, called an application scheme.

An application scheme makes it possible to set the main object (client or contract) for which the application is intended and a hierarchy of related objects.

A scheme consists of sections, contains a main section (for entering data about the main object) and set of subordinate sections organised in a hierarchy. Each section is used to enter information about a system object (client, contract, address, etc.).

Example of a hierarchical structure of objects:

- Client record (main object).
 - Contract.

- Address.
- Card contract.

2 Service Catalogues

Service catalogues are used to set lists of objects for which applications are intended, permitted actions, etc.

Service catalogues include:

- Catalogue of application objects.
- Catalogue of actions with objects.
- Classifiers of application additional information.
- Product codes.

2.1 Catalogue of Objects

The catalogue of objects (Advanced Applications R2 → Setup → Appl Object Type) contains a list of system objects with which the Advanced Applications module can operate (see [figure](#)).

Appl Object Type					<< < > >>	1 of 41	✕
	Object For Type	Product Category	Code	Name	Used Objects		
→			CustomerClassifier	Customer Classifier	CustomerClassifier;		
			Contract	Contract	Contract;Limit;Card;Address;		
			ClientContract	Client and Contract	Client;Contract;Limit;Card;Address;		
			Client	Client	Client;		
	Officer Group		OfficerGroup	Officer Group	OfficerGroup;		
	Officer		Officer	Officer	Officer;		
	Contract		Usage	Usage Limiter	Usage;		
	Contract		Tariff	Tariff	Tariff;		
	Contract	Issuing	StopList	Stop List			
	Contract		Status	Status	Status;		
	Contract		Relation	Relation	Relation;		
	Contract		Preferred	Preferred	Preferred;		
	Contract		Order	Standing Order	Order;		
	Contract	Issuing	Limit	Credit Limit	Limit;		
	Contract	Issuing	InstalmentLimit	Instalment Limit			
	Contract	Issuing	Instalment	Instalment			
	Contract		Event	Event	Event;		
	Contract	Acquiring	DeviceParms	Device Parameters			
	Contract		ContractParameter	Contract Parameter	CustomerParameter;		
	Contract	Issuing	ContractMemo	Contract Memo			
	Contract		ContractGroupAffiliation	Contract Group Affiliation	ContractGroupAffiliation;		

Catalogue of system objects

In the *Name* column, all Advanced Applications module objects are listed for which applications to create or change object parameters can be registered.

The *Object For Type* field specifies the system object to which the corresponding Advanced Applications module object belongs. For example, the *ContractAddress* object belongs to a contract

registered in the system. The value in this field is used for filtering module objects when working with forms.

The *Product Category* field shows the category to which a given object belongs (Issuing or Acquiring).

The *Code* field contains the object identifier code (usually the value in this field corresponds to the value in the *Name* field).

The *Used Objects* field contains a list of objects actions on which can be provided for by an application. List elements are delimited by semi-colons.



To work with volume-based tariffs, add a record to the "Appl Object Type" form (or make sure the record is present), with the following field values: *Object For Type* = "Contract", *Code* = "VolumeBasedTariff", *Name* = "Volume Based Tariff", *Used objects* = "VolumeBasedTariff;".

Use of volume-based tariffs requires an additional agreement with OpenWay.



Records in the "Appl Object Type" form can be edited by OpenWay representatives only. This rule does not apply to adding settings for working with volume-based tariffs.

2.2 Catalogue of Actions with Objects

In the catalogue of actions with objects (Advanced Applications R2 → Setup → Appl Action Type) all actions are listed that can be executed with system objects using the Advanced Applications module.

Appl Action Type			<< < > >>	1 of 90	X
	Use for Object Type	Code	Name		
➔	AddInfo	Add	Add		
	AddInfo	Update	Update		
	Card	Add	Add		
	Card	Lock	Lock		
	Card	Unlock	Unlock		
	Card	Update	Update		
	Client	Add	Add		
	Client	AddOrUpdate	Add Or Update		
	Client	Update	Update		
	Client Add Info	Add	Add		
	Client Add Info	AddOrUpdate	Add Or Update		
	Client Address	Add	Add		
	Client Address	AddOrUpdate	Add Or Update		
	Client Address	Disable	Disable		
	Client Address	Enable	Enable		
	Client Affiliation	AddOrUpdate	Add Or Update		
	Client Affiliation	Disable	Disable		
	Client Auth Method	AddOrUpdate	Add Or Update		
	Client Auth Method	Disable	Disable		
	Client Classifier	AddOrUpdate	Add Or Update		
	Client Classifier	Disable	Disable		
	Client and Contract	Add	Add		
	Client Counterparty	AddOrUpdate	Add Or Update		
	Client Parameter	AddOrUpdate	Add Or Update		
	Client Parameter	Disable	Disable		
	Contract	Add	Add		
	Contract	Update	Update		
	Contract Add Info	Add	Add		
	Contract Add Info	AddOrUpdate	Add Or Update		
	Contract Address	Add	Add		
Ins Del Query					

Actions with system objects

In the *Use for Object Type* column, all objects from the catalogue of Advanced Applications objects are listed (see [figure](#)).

Each row of the list corresponds to one action than can be executed with the corresponding object. The action is shown in the *Name* field. The *Code* field specifies an action code.



To work with volume-based tariffs, add records to the "Appl Action Type" form (or make sure the records are present), with the following field values:

- Use for Object Type = "Volume Based Tariff", Code = "Add", Name = "Add".
- Use for Object Type = "Volume Based Tariff", Code = "Disable", Name = "Disable".
- Use for Object Type = "Volume Based Tariff", Code = "CancelFutureChanges", Name = "CancelFutureChanges".

Use of volume-based tariffs requires an additional agreement with OpenWay.



Records in the "Appl Action Type" form can be edited by OpenWay representatives only. This rule does not apply to adding settings for working with volume-based tariffs.

2.3 Application Additional Information Classifiers (Application Info Types)

For each application, additional information can be specified in the special APPL_INFO table. This information can be used to execute custom check or approve procedures.

Additional information is stored in the APPL_INFO table with the data type specified. Additional information data types are additional information classifiers (Application Info Type).

Data is entered in the APPL_INFO table using custom forms. These forms are configured using data filtering according to the value of the Application Info Type code (see the section "Form Builder Window: the "Fields" Tab" in the document "DB Manager. Form Builder").

Properties of application additional information types are specified in the "Application Info Type" form (Advanced Applications R2 → Setup → Application Info Types):

Application Info Types								<< < > >>		1 of 61	X
Name	Code	Initial Count	Max Count	Group Code	Is Manual	For Whom	Is Ready				
Reinstated Effective Date	MWF_EFF_DATE_R	0	0	X		Merchant Management	Ready				
Role Type Code	MWF_ROLE_CODE	0	0	X		Merchant Management	Ready				
Cancellation Effective Date	MWF_EFF_DATE_C	0	0	J		Merchant Management	Ready				
Reason Code	MWF_REASON_CODE_J	0	0	J		Merchant Management	Ready				
Agent Code	MWF_ACO_CODE	0	0	J		Merchant Management	Ready				
MasterCard TCC Override	MC_TCC_OVERRIDE	0	0	E		Device Parm to Doc	Ready				
VISA Merchant Verification Value	VISA_MVV	0	0	V		Device Parm to Doc	Ready				
Contactless Payment Acceptor	MWF_CTLIS_IND	0	0	X		Merchant Management	Ready				
Card Acceptor Full Recourse Status	MWF_FULL_RECOURSE_IND	0	0	X		Merchant Management	Ready				
Card Acceptor High Risk Indicator	MWF_HIGH_RISK_IND	0	0	X		Merchant Management	Ready				
Key Account Indicator	MWF_KEY_ACC_IND	0	0	X		Merchant Management	Ready				
Visa Card Indicator	MWF_VISA_CARD_IND	0	0	X		Merchant Management	Ready				
MasterCard Indicator	MWF_MC_CARD_IND	0	0	X		Merchant Management	Ready				
Other Card Indicator	MWF_OTHER_CARD_IND	0	0	X		Merchant Management	Ready				
International Establishment Code	DCL_INTES_CODE	0	0	C		Device Parm to Doc	Ready				
MasterCard Assigned ID	MC_ASSIGNED_ID	0	0	E		Device Parm to Doc	Ready				
MasterCard Mobile Phone Service Provider Name	MC_MOBILE_PROVIDER_NAME	0	0	E		Device Parm to Doc	Ready				
VISA Mobile Phone Service Provider Name	VISA_MOBILE_PROVIDER_NAME	0	0	V		Device Parm to Doc	Ready				
P2P Client Info	P2P_CLIENT	0	0			Client	Ready				
VISA Large Merchant	VISA_LM	0	0	V		Device Parm to Doc	Ready				
Use Payment Facilitator	USE_PAYM_FCLTR	0	0	E		Device Parm Ctrl	Ready				

Form for specifying additional info type properties

The form's fields:

- *Name* – classifier name.
- *Code* – classifier code. This value is used when creating an additional information input form. The combination of the *Code* and *Group Code* field values must be unique.
- *Initial Count* – the number of records in the application additional information table (APPL_INFO) that will be created when the form is opened for the first time.
- *Max Count* – the maximum number of records for one application in the additional information table for this classifier type. This value can be controlled with custom procedures.
- *Group Code* – the classifier group code; this value is used to group application additional information.
- *Is Manual* – flag to indicate ("Yes" value) that the corresponding record in the APPL_INFO table is created by the user. This field can be used to sort records in the APPL_INFO table.
- *For Whom* – the field specifies the object or process for which additional information is intended:
 - Client (C) – client parameters.
 - Contract (A) – contract parameters.
 - Merchant Management (M) – merchant management.
 - Device Parm to Doc (D) – device parameters (tags) that must be saved into the *Add_Info* field of documents.
 - Device Parm Ctrl (S) – device parameters that affect document processing.

The [Check] button is used to call a procedure that validates the selected additional information data type.

Records are created in the additional information table and the correspondence of the number of records to set limits is checked according to standard system procedures.

2.4 Product Codes

When entering an application to change or add a contract record in the Advanced Applications module, it is possible to specify the Product for the contract by selecting a Product code.

A Product is selected by specifying its code when there are a large number of Products registered in the system. When selecting a Product by code, only those Products for which a code is specified will be shown in the list, meaning Product codes are used for filtering the Product list.

To select a Product by code, codes must have been set earlier for Products used when entering applications (Advanced Applications R2 → Setup → Appl Product Code). The list of Product codes is organized as a hierarchy with three (or fewer) levels.

The list of Product codes is configured in the "Appl Product Codes" form (see [Figure 4](#)).

Appl Product Codes				<< < > >>		1 of 7		X
	Handbook	Name	Code	Parent Product Code 1	Parent Product Code 2			
→	Product Code 1	Product 1	PR1					
	Product Code 2	Product 1.1	PR1.1	Product 1				
	Product Code 3	Product 1.1.1	PR1.1.1	Product 1	Product 1.1			
	Product Code 2	Product 1.2	PR1.2	Product 1				
	Product Code 1	Product 2	PR2					
	Product Code 2	Product 2.1	PR2.1	Product 2				
	Product Code 3	Product 2.1.1	PR2.1.1	Product 2	Product 2.1			
Ins Del Query								

Figure 4. Product codes

Fill in the following fields of this form:

- *Handbook* – drop-down list to select the Product code hierarchy level; this field can have one of the following values:
- "Product Code 1" – the top level of the hierarchy. For a Product code with this value in the *Handbook* field:
- Parent Product codes are not specified (values in the *Parent Product Code 1* and *Parent Product Code 2*) fields.
- The value in the *Code* field must be the same as the value of the *Product Code* field set for the Product when configuring additional parameters (Full → Configuration Setup → Products → Product Definition → Products).
- "Product Code 2" – the second level of the hierarchy. For a Product code with this value in the *Handbook* field:
- The parent Product code (top level of the hierarchy) is specified in the *Parent Product Code 1* field.
- The value in the *Code* field must be the same as the value of the *Code 2* field set for the Product when configuring additional parameters (Full → Configuration Setup → Products → Product Definition → Products).
- "Product Code 3" – the third (lowest) level of the hierarchy. For a Product code with this value in the *Handbook* field:
- The parent Product code (top level of the hierarchy) is specified in the *Parent Product Code 1* field.
- The parent Product code (second level of the hierarchy) is specified in the *Parent Product Code 2* field.
- The value in the *Code* field must be the same as the value of the *Code 3* field set for the Product when configuring additional parameters (Full → Configuration Setup → Products → Product Definition → Products).
- *Name* – Product code name.
- *Code* – Product code.
- *Parent Product Code 1* – field for selecting the parent Product code (top level of the hierarchy).
- *Parent Product Code 2* – field for selecting the Parent product code (second level of the hierarchy).

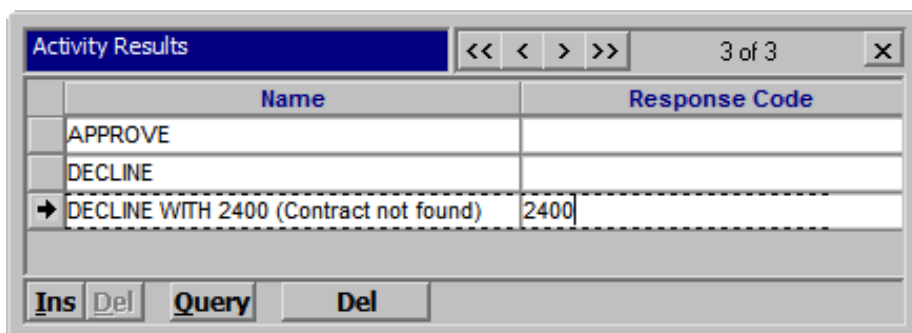
3 Advanced Applications Module Configurations

When setting up the Advanced Applications module, it is possible to set a list of users and their privileges, and application workflow strategy.

3.1 Service Settings

In the current version of the module, the catalogue of workflow activity results is a service setting.

The catalogue of workflow activity results is configured in the "Activity Results" form (Advanced Applications R2 → Setup → Workflow Configuration → Activity Results) for all domains (see the section "[Domains](#)").



Name	Response Code
APPROVE	
DECLINE	
DECLINE WITH 2400 (Contract not found)	2400

Activity results

The form's fields:

- *Name* – name of the application processing result.
- *Response Code* – error code for an application processing result.

The application processing result can later be linked to a particular action (see the section "[Workflow Activities](#)").

For actions whose execution results is a particular error code (*Response Code*), a special step can be set up when configuring rules in workflow strategies (see the section "[Workflow Strategies](#)").

Existing error codes are described in the section "Response File Error Messages" of the document "XML Applications (R2)".

By default, the "Activity Results" form contains two standard records of application processing results. These records may not be edited:

- "Decline" – the action was executed with an error, the application was declined.
- "Approve" – the action was executed successfully, the application was approved.

Activity results are used when creating rules required, in particular, to transfer applications from one step to another (see "[Selecting a Strategy](#)").

3.2 User Administration

Module users work within groups registered for a domain (see "Domains").

Users are distinguished by privileges to execute activities with applications. A set of user privileges (list of permitted activities) is determined by the role assigned to the user.

User groups are registered within domains using the "User Groups" form (Advanced Applications R2 → Workflow Configuration → User Management → User Groups).

Domain	Name
Application WorkFlow	Application Clerk

Ins Del Query

User groups

User roles are registered and configured using the "User Roles" form (Advanced Applications R2 → Workflow Configuration → User Management → User Roles).

Domain	Name	Permit All
Application WorkFlow	Application Clerk	Yes

Ins Del Query Del Activities

Activities for Application Clerk

Activity
Approve Application

Ins Del Query

Configuring user roles

Roles are registered within a domain.

When the value of the *Permit All* field is set to "Yes", users with the given role may execute all activities with applications.

If the "No" value is set in this field, the role may be permitted to execute certain activities.

To assign permitted activities to a role, click the [Activities] button and in the "Activities for <name of role>" form, use the drop-down list to add the desired activities.

Users are registered in the "Users" form (Advanced Applications R2 → Workflow Configuration → User Management → Users).

Users						
Domain	Group	Name	Connected As	Is Active	Is Ready	
Application WorkFlow	Application Clerk	Application User	SUPERUSER	Yes	Ready	

Roles for Application User	
Role	Is Ready
Application Clerk	Ready

User registration

To register a user, do as follows:

- Specify the domain name in the *Domain* field.
- Specify the group in which the user will be included in the *Group* field.
- In the *Connected As* field, specify the DB Manager user name under which the Advanced Applications module user will connect to the Way4 database.
- To activate the user, set "Yes" in the *Is Active* field. When necessary, the field can be used to deactivate the user.
- In a child form opened by clicking the [Roles] button, assign roles to the user. Each user can be assigned several roles.
- To approve changes, click the [Approve] button. If changes are approved successfully, the *Is Ready* field will be set to "Yes".

3.3 Domains

The use of domains in the Advanced Applications module ensures application grouping by financial institutions.

Application workflow strategies are registered and used within a domain.

Domains are registered and configured in the "Domains" form (Advanced Applications R2 → Workflow Configuration → Domains).

Domains	
Priority	Name
0	Application WorkFlow

Financial Inst for Application WorkFlow	
Financial Institution	

Form for registering and configuring domains

In the *Priority* field, specify the value determining the procedure for selecting domains (the lower the value, the earlier the domain will be checked).

Clicking the [Financial Inst] button opens the form to specify the financial institution to which the domain belongs.

If a domain for the required financial institution is not found during selection, a domain will be chosen for which no financial institution is specified (if such a domain is configured).

3.4 Application Parameters

Application parameters are used by the module when selecting a workflow strategy and assigning a user (see "[Selecting a Strategy](#)").

Application parameters must have a value from the list of possible values. A value is assigned to a parameter after comparing the result of estimating (calculating) its value with the list of possible values.

Parameters are registered and configured in the "Parameters" form (Advanced Applications R2 → Workflow Configuration → Parameters).

Domain	Seq Num	Name	Code	Parameter Source	Source Data	Appl Info Type	Currency	Default Value	Unknown Value	Calculate	Is Ready
Application Workflow	1	Creation Type	CREATION_TYPE	Client/Contract/Application	%CREATION_TYPE%			Manual Entry	Manual Entry	Yes	Ready
Application Workflow	16	Usage	USAGE	Client/Contract/Application	%USAGE_CODE%					Yes	Ready
Application Workflow	2	Application Type	APPLICATION_TYPE	Client/Contract/Application	%APPLICATION_TYPE%					No	Ready
Application Workflow	3	Product Group	PGROUP	Client/Contract/Application	%PRODUCT_GROUP%					No	Not Ready
Application Workflow	4	Financial Institute	FI	Client/Contract/Application	%FI_%					No	Ready
Application Workflow	5	Client Category	CCAT	Client/Contract/Application	%CCAT%					No	Ready
Application Workflow	6	Branch	BRANCH	Client/Contract/Application	%BRANCH%					No	Ready
Application Workflow	7	Product Category	PCAT	Client/Contract/Application	%PCAT%					No	Ready
Application Workflow	8	Appl Object Type	OBJ_TYPE	Client/Contract/Application	%OBJECT_TYPE%					No	Ready
Application Workflow	9	Appl Action Type	ACT_TYPE	Client/Contract/Application	%ACTION_TYPE%					No	Ready
Application Workflow	10	Credit Limit	CR_LIMIT	Client/Contract/Application	%CREDIT_LIMIT%					No	Ready
Application Workflow	11	Address ZIP	ZIP	Client/Contract/Application	%ADDRESS_ZIP%					No	Ready
Application Workflow	12	Product	PR_CODE_1	Client/Contract/Application	%PRODUCT_CODE_1%					No	Ready
Application Workflow	13	Client Age	AGE	Client/Contract/Application	%AGE%					No	Ready
Application Workflow	14	Appl Source Code	SOURCE	Client/Contract/Application	%SOURCE_CODE%					No	Ready

Priority	Name	Code	Source Result	From Amount	To Amount	Program Code	Is Ready
0	Acquiring Cash POS and ATMs	ACDCASH	ACDCASH	0.00	0.00		Not Ready
1	Issuing Debit	ISSDEB	ISSDEB	0.00	0.00		Ready
2	Acquiring Retail	ACQRET	ACQRET	0.00	0.00		Ready

Configuring application parameters

The form's fields:

- *Seq Num* – record 's ordinal number. The field is used to sort values in this form.
- *Parameter Source* – object containing the parameter:
 - "Client/Contract/Application" – the application.
 - "Application Info" – the record in the APPL_INFO table of additional information.
 - "Application Add Data" – the ADD_DATA field of the application.
- *Source Data* – type of data received from the object:
 - The following data types may be specified for application fields:
 - ENTRY – manual input (screen forms).
 - IMPORT – import from a file using an import pipe.

- WS – entry using web services (automatic creation of applications when working with customer service workbench, merchant service workbench and merchant portal).
- SPLIT – application created as subordinate, further subject to deferred acceptance (see the section "Accepting Applications" of the document "Advanced Applications R2").
- *Program Code* = "INCLUDE" – the parameter value will be returned if the value specified in the *Source Result* field is included in the list of classifier types.
- *Program Code* = "EXCLUDE" – the parameter value will be returned if the value specified in the *Source Result* field is not included in the list of classifier types.
- %APPLICATION_TYPE% – application type.
- %SOURCE_CODE% – code of the system in which the application was created.
- %DUPLICATE_COUNT% – number of duplicate applications.
- %CREATION_TYPE% – application creation method. Possible values:
- %OBJECT_TYPE% – application object type (see "[Catalogue of Objects](#)").
- %ACTION_TYPE% – action with an application object (see "[Catalogue of Actions with Objects](#)").
- %F_I% – client financial institution.
- %BRANCH% – client branch.
- %CCAT% – client category.
- %CLIENT_TYPE% – client type.
- %CLASSIFIER_TYPE% – type of client/contract classifier. When configuring parameter values, note that if an application contains several sections for working with classifiers, all classifier type codes will be grouped in a list (with commas to separate them) and compared with the value specified in the *Source Result* field (see the section "Case Parameters" of the document "Way4™ Case Management Configuration"). To determine the presence of sections with the specified classifier type in an application, it is recommended to specify the following values in the *Program Code* field:
 - %PCAT% – contract Product category.
 - %LANGUAGE% – client language, set in the application.
 - %AGE% – client age calculated according to data of birth specified in the application.
 - %CREDIT_LIMIT% – credit limit.
 - %ADDRESS_ZIP% – postal (ZIP) code (part of address) specified in the application.
 - %PRODUCT_CODE_1%, %PRODUCT_CODE_2%, %PRODUCT_CODE_3% – Product code.
 - %PRODUCT_GROUP% – code of the group to which the Product for the contract specified in the application belongs.

- For records in the APPL_INFO table of additional information: %ADD_INFO_01% – the value from the specified field of the APPL_INFO table of additional information with the data type specified in the *Appl Info Type* field.
- For the ADD_DATA field of the application: %TAG NAME%. Moreover, in the "Values for <parameter name>" form, a record must be created for each of the possible tag values. This data type may be used to select a workflow strategy.
- *Calculate* – a value in this field specifies whether this parameter will be calculated for the application.
- *Default Value* – parameter value used by default when a calculated value for the parameter could not be obtained.
- *Unknown Value* – parameter value if the resulting value could not be matched with any possible value.

Possible parameter values are set in the "Values for <name of parameter>" form, opened by clicking the [Values] button. This form is used for parameters with text or numeric values.

The *Source Result* field of the "Values for <name of parameter>" contains text with which the parameter value is compared. The *From Amount* and *To Amount* fields specify the lower and upper values of the range of numeric parameter values.

The following procedure is used to determine parameter values:

- The estimation of a text parameter value is compared with the value in the *Source Result* field. If correspondence is determined, the parameter is assigned the value contained in the *Name* field of the string found.
- The estimation of a numeric parameter value is compared with the value in the *From Amount* and *To Amount* fields. If a value is found to fall in this range, the parameter is assigned the value contained in the *Name* field of the string found.
- A search is made in the order determined by the value of the *Priority* field (the lower the value, the earlier the match is checked).
- If no correspondence with the estimated (calculated) value was found in the list of possible parameter values, the parameter is assigned the value specified in the *Unknown Value* field.

After configuration is completed, confirm the changes by clicking the [Actions...] button and selecting the "Approve" or "Approve All" item from the context menu. In the "Date Start" form that opened (see [figure](#)), set the date on which changes become effective (current or future date can be specified), then click the [Proceed] button.



Setting the date on which changes become effective

Additional information about the results of approval can be obtained after selection in the user menu item "Full → Process Log → Process Log".

To delete a parameter from the list, in the strategy schemes form click the [Actions...] button and select "Del" from the context menu.

The [History] button in the "Parameters" form (see [figure](#)) opens the "History for <name of parameter>" form (see [figure](#)) that contains all previous versions of the selected parameter and the parameter's version that is currently used by Way4.

The *Date From* and *Date To* fields determine the period for which the parameter version is effective:

- *Date From* – date from which Way4 will use this version of the parameter. This date is specified when changes are applied.
- *Date To* – date from which the next version of the parameter becomes effective. If the field contains "00/00/0000", this means the next version has not been created yet.

To view values of the parameter version, select the required record in the "History for <name of parameter>" form and click the [Values] button (see [figure](#)).

Date From	Date To	Domain	Seq Num	Name	Code	Parameter Source	Source Data	Appl Info Type	Currency	Default Value	Unknown Value	Calculate
12/02/2018	00/00/0000	Application Workflow	1	Creation Type	CREATION_TYPE	Client/Contract/Application	%CREATION_TYPE%			Manual Entry	Manual Entry	Yes
12/02/2018	12/02/2018	Application Workflow	1	Creation Type1	CREATION_TYPE	Client/Contract/Application	%CREATION_TYPE%			Manual Entry	Manual Entry	Yes
01/01/2000	12/02/2018	Application Workflow	1	Creation Type	CREATION_TYPE	Client/Contract/Application	%CREATION_TYPE%			Manual Entry	Manual Entry	Yes

Priority	Name	Code	Source Result	From Amount	To Amount	Program Code	Is Ready
0	Obsolete Application	OBSOLETE	OBSOLETE	0,00	0,00		Ready
201	Manual Entry	ENTRY	ENTRY	0,00	0,00		Ready
202	Application Split	SPLIT	SPLIT	0,00	0,00		Ready
203	WebService	WS	WS	0,00	0,00		Ready
205	Online Load	ONLINELOAD	ONLINELOAD	0,00	0,00		Ready
206	File Import	IMPORT	IMPORT	0,00	0,00		Ready

History of changes to a parameter

3.5 Workflow Activities

To configure the catalogue of activities that can be executed by a user or the system during application workflow, use the "Activities" form (Advanced Applications R2 → Setup → Workflow Configuration → Activities).

Activities								<< >>		1 of 9	X
Domain	Name	IDT	Has Result	Is System	WF Operation	Program Code	UI Action	Is Ready			
Application Workflow	Accept Application	ACCEPT	Yes	No	To Closed	APPL_API=ACCEPT;		Ready			
Application Workflow	Approve Application	APPROVE	Yes	No	To Open	APPL_API=APPROVE;		Ready			
Application Workflow	Capture and Edit Application	CAPTURE_EDIT	No	No		APPL_API=CAPTURE;	W4W_SHOW_BUTTON=Y;CAPT	Ready			
Application Workflow	Capture and Edit Application 2	CAPTURE_EDIT_2	No	No		APPL_API=CAPTURE;	W4W_SHOW_BUTTON=Y;CAPT	Ready			
Application Workflow	Check Application	CHECK	No	No	No Change	APPL_API=CHECK;		Ready			
Application Workflow	Decline Application	DECLINE	No	No	To Closed	APPL_API=DECLINE;		Ready			
Application Workflow	Export Application	EXPORT	No	No	No Change		Export Action	Ready			
Application Workflow	Forward Application	FORWARD	No	No	No Change		Forward Action	Ready			
Application Workflow	System Decline Application	SYS_DECLINE	No	Yes	To Closed	APPL_API=DECLINE;		Ready			

Ins	Del	Query	Actions...	Results	History
-----	-----	-------	------------	---------	---------

Results for Accept Application								<< >>		1 of 4	b X
Result											
APPROVE											
APPROVE											
DECLINE											
SECONDARY_NODE_ACCEPT											

Ins	Del	Query	Del
-----	-----	-------	-----

Configuring workflow activities

The parameters of activities that can be executed when an application is processed are specified in this form.

The *Has Result* field specifies whether the respective activity is characterized by a result. If the "Yes" value is set in this field, the [Results] button will become available for the activity. This button opens the form for specifying possible results of the activity's execution. Results of activity execution are selected from the "Activity Results" catalogue (see the section "[Service Settings](#)").

The *Is System* field specifies if an activity is executed by the system or by a user. For system activities (the "Yes" value), before executing an activity, user privileges for executing this activity are not checked.

The *WF Operation* field defines the status of the application workflow process after an activity has been performed:

- "To Closed" – close the application workflow process.
- "To Open" – reopen the application workflow process.
- "No Change" – do not change the status of the application workflow process.

In a standard strategy, "To Closed" is set for activities to accept or decline applications ("Accept Application", "Decline Application" and "System Decline Application"). "To Open" is set for activities to approve or edit an application ("Approve Application" and "Capture and Edit Application").

After changing application workflow settings, it is recommended to close all processed applications as a group (applications with the "Posted" and "Declined" values). To do so, use the menu item "Advanced Applications R2 → Application Processing → Close Posted Applications".

To improve performance, it is recommended to run this process with the "PARALLEL" parameter (see the document "Running Processes in Parallel").

The *Program Code* field contains the code for calling an API procedure. The value in this field may only be modified by OpenWay representatives.

Depending on the value specified in the *Program Code* field, non-system activities (with the "No" value in the *Is System* field) may be available to the user when working with various elements of the Advanced Applications module user interface.

The *UI Action* field makes it possible to specify the button (or other interface element) that will call execution of this action. This field may contain the following values:

- "Capture Action" – makes it possible to call the specified activity from Advanced Applications R2 module screen forms by clicking the [Actions...] button, and subsequently selecting the "Capture" menu item (or by clicking the [Capture] button).
- "Export Action" – makes it possible to call the specified activity by selecting the user menu item "Advanced Applications R2 → Application Processing → XML Applications Export".
- "Forward Action" – makes it possible to call an activity from Advanced Applications R2 screen forms by selecting the "Forward" context menu item called by clicking the [Actions...] button. When this item is selected, a window opens for forwarding the application. This window contains the list of all activities available for this strategy step for which the "Forward Action" value was specified in the *UI Action* field during configuration.

Usually, for an activity record, it is recommended to specify a value either in the *Program Code* field or in the *UI Action* field.

The table below shows the correspondence between values in the *Program Code* or *UI Action* fields of activity records and interface elements in which these activities may be available.

Availability of activities in the Advanced Applications R2 module user interface depending on values in the Program Code or UI Action field

Value in the <i>Program Code</i> or <i>UI Action</i> field	Name of the workflow activity	Interface element in which the activity is available	User menu item or button
<i>Program Code</i> = "APPL_API= ACCEPT"	Accept an application (see the section "Accepting Applications" of the document "Advanced Applications R2")	User menu.	Accept processed issuing module applications: "Advanced Applications R2 → Application Processing → Issuing Applications – Accept". Accept processed acquiring module applications: "Advanced Applications R2 → Application Processing → Acquiring Applications – Accept".

Value in the <i>Program Code</i> or <i>UI Action</i> field	Name of the workflow activity	Interface element in which the activity is available	User menu item or button
<i>Program Code</i> = "APPL_API= APPROVE"	Approve an application (see the section "Approving Applications" of the document "Advanced Applications R2")	Application input form. Form for manually processing applications.	The "Approve" item of the context menu opened by clicking the [Actions...] button.
<i>Program Code</i> = "APPL_API= CHECK"	Check an application (see the section "Checking Applications" of the document "Advanced Applications R2")	Application input form. Form for manually processing applications.	The "Check" item of the context menu opened by clicking the [Actions...] button.
<i>Program Code</i> = "APPL_API= DECLINE"	Decline an application (see the section "Declining Applications" of the document "Advanced Applications R2")	Form for manually processing applications.	The "Decline" item of the context menu opened by clicking the [Actions...] button.
<i>UI Action</i> = "Forward Action"	Forward an application (see the section "Forwarding Applications" of the document "Advanced Applications R2")	Form for manually processing applications.	<p>The "Forward" item of the context menu opened by clicking the [Actions...] button.</p> <p>When this item is selected, a window opens for forwarding the application. This window contains the list of all actions available for this strategy step for which the "Forward Action" value was specified in the <i>UI Action</i> field during configuration.</p>

Value in the <i>Program Code</i> or <i>UI Action</i> field	Name of the workflow activity	Interface element in which the activity is available	User menu item or button
<i>UI Action</i> = "Capture Action"	"Captures" an application and opens the application form in editing mode (see the sections "Processing Approved Applications" and "Processing Declined Applications" of the document "Advanced Applications R2")	The list of imported applications, approved applications and the list of declined applications.	The [Actions...] button, with subsequent selection of the "Capture" menu item, or the [Capture] button.
<i>UI Action</i> = "Export Action"	Export an application (see the section "Exporting Applications" of the document "Advanced Applications R2")	User menu.	"Advanced Applications R2 → Application Processing → XML Applications Export".



The activities listed above are available in the specified screen forms if they have been added to the list of activities permitted for the strategy according to which the application is processed, or to the list of activities available for a specific strategy step (see the section "[Workflow Strategies](#)").

After configuration is completed, confirm the changes by clicking the [Actions...] button and selecting "Approve" from the context menu. In the "Date Start" form that opened (see [figure](#)), set the date on which changes become effective (current or future date can be specified), then click the [Proceed] button.

Additional information about the results of approval can be obtained after selection in the user menu item "Full → Process Log → Process Log".

To delete activities from the list, click the [Actions...] button and select "Delete" from the context menu.

The [History] button in the "Activities" form (see [figure](#)) opens the "History for <name of activity>" form (see [figure](#)) that contains all previous versions of the selected workflow activity and the activity's version that is currently used by Way4.

The *Date From* and *Date To* fields determine the period for which the workflow activity version is effective:

- *Date From* – date from which Way4 will use this version of the activity. This date is specified when changes are applied.
- *Date To* – date from which the next version of the activity becomes effective. If the field contains "00/00/0000", this means the next version has not been created yet.

To view possible results of activity execution that are specified for a specific version, select the required record in the "History for <name of activity>" form and click the [Results] button (see [figure](#)).

History for Accept Application										<< < > >>		1 of 2		b x	
Date From	Date To	Domain	Name	IDT	Has Result	Is System	WF Operation	Program Code	UI Action						
01/01/2017	00/00/0000	Application WorkFlow	Accept Application	ACCEPT	Yes	No	To Closed	APPL_API=ACCEPT;							
01/01/2000	01/01/2017	Application WorkFlow	Accept Application	ACCEPT	Yes	No	To Closed	APPL_API=ACCEPT;							

Ins Del Query Results

Results for Accept Application										<< < > >>		1 of 3		b x	
Result															
APPROVE															
DECLINE															
SECONDARY_NODE_ACCEPT															

Ins Del Query

Workflow activity history of changes

3.6 Workflow Strategies

Workflow strategies are registered within domains (see the section "[Domains](#)") and are grouped in schemes related to, for example, issuing, acquiring, etc.

To configure workflow strategies, select the "Advanced Applications R2 → Workflow Configuration → Strategy Schemes" menu item. This will open the "Strategy Schemes" form together with a group of child forms.

Strategy Schemes					<< < > >>		1 of 2		x	
Domain	Name	Code	Is Ready							
Application WorkFlow	Application Workflow	APPL_ALL_WF	Ready							
Application WorkFlow	Application Workflow _Copy	APPL_ALL_WF	Ready							

Ins Del Query Actions... Strategies History

Strategies for Application Workflow				<< < > >>		1 of 4		b x	
Name	Code	Is Ready							
Manual Entry Application Strategy	ENTRYSIMPLE	Ready							
File Import Application Strategy	IMPORTSIMPLE	Ready							
Online Load Application Strategy	LOADSIMPLE	Ready							
WS Entry Application Strategy	WSENTRYSIMPLE	Ready							

Ins Del Query Actions... Steps Activities Rules

Steps for Manual Entry Application Strategy				<< < > >>		1 of 4		b x	
Seq Num	Name	Code	Is Ready						
0	Manual entry step	ENTRY	Ready						
1	Accepting Step	ACCEPT	Ready						
2	Declining Step	DECLINE	Ready						
3	Closing Step	CLOSE	Ready						

Ins Del Query Del Activities

General Activities for Manual Entry		<< < > >>		1 of 2		b x	
Activity	Is Ready						
System Decline Application	Ready						
Check Application	Ready						

Ins Del Query Del

Special Activities for Manual entry step		<< < > >>		1 of 4		b x	
Activity	Is Ready						
Approve Application	Ready						
Capture and Edit Application	Ready						
Decline Application	Ready						
Check Application	Ready						

Ins Del Query Del

Form for configuring workflow strategies

In the "Strategy Schemes" form, schemes used to group workflow strategies are registered within domains.

Strategies included in a scheme are registered in the child form "Strategies for <name of scheme>", opened by clicking the [Strategies] button in the parent form.

The following are configured for a strategy:

- Activities that can be executed during workflow within the strategy (executed in all steps of the strategy).
- Workflow steps.

To configure activities, click the [Activities] button in the strategies form to open the child form "General Activities for <name of strategy>". Activities that can be added to a strategy are selected from a list created earlier (see "[Workflow Activities](#)").

To create a copy of the strategy, in the strategies form click the [Actions...] button and select "Duplicate" from the context menu.

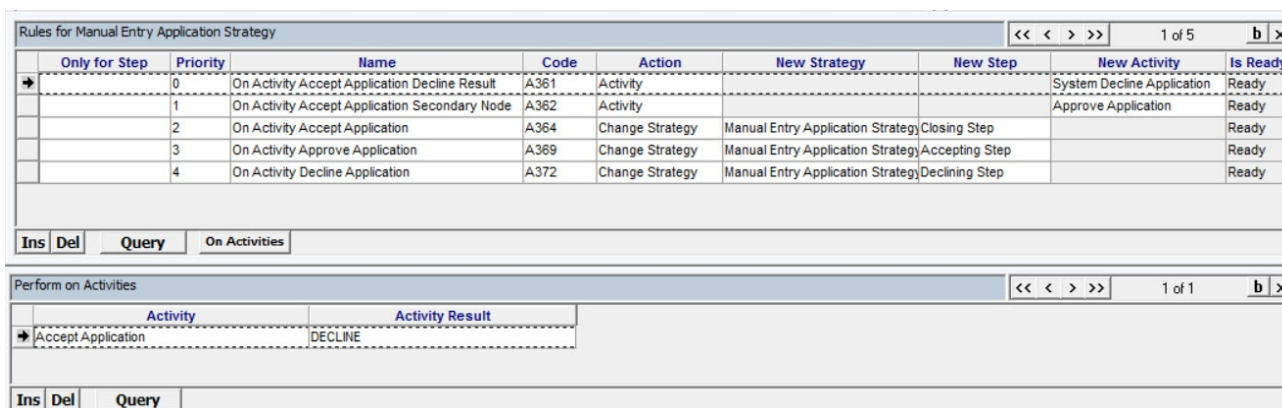
To delete a strategy from the list, in the strategies form, click the [Actions...] button and select "Delete" from the context menu.

To configure workflow steps, click the [Steps] button in the strategies form to open the child form "Steps for <name of strategy>". In this form, the user creates the sequence of workflow steps, specifying the sequence number of the step in the *Seq Num* field.

For each registered step, an individual set of activities that can be executed when the application is located at this step may be configured. This setting is made in the same way as for strategies, in the "Special Activities for <name of step>" form opened by clicking the "Activities" button.

An application is transferred from one step to another within a workflow strategy, or transferred to another strategy when rules configured for the strategy or step are fulfilled.

To configure rules, for the selected strategy in the form "Strategies for <name of scheme>", click the [Rules] button to open the child form "Rules for <name of strategy>" (see [figure](#)).



Only for Step	Priority	Name	Code	Action	New Strategy	New Step	New Activity	Is Ready
→	0	On Activity Accept Application Decline Result	A361	Activity			System Decline Application	Ready
	1	On Activity Accept Application Secondary Node	A362	Activity			Approve Application	Ready
	2	On Activity Accept Application	A364	Change Strategy	Manual Entry Application Strategy	Closing Step		Ready
	3	On Activity Approve Application	A369	Change Strategy	Manual Entry Application Strategy	Accepting Step		Ready
	4	On Activity Decline Application	A372	Change Strategy	Manual Entry Application Strategy	Declining Step		Ready

Ins Del Query On Activities

Activity	Activity Result
→ Accept Application	DECLINE

Ins Del Query

Configuring rules

In the Advanced Applications module a rule is a condition and the result of the condition being met.

A rule's condition is the execution of a workflow activity, including with a specified result. Conditions for configurable rules are set in the child form "Perform on Activities", opened by clicking the [On Activities] button.

In this form's *Activity* field, specify the activities whose execution is a condition for the rule to be applied. In the *Activity Result* field, the result of executing the activity may be specified. If not, execution of the activity with any result will be considered a condition for applying the rule.

When configuring conditions, workflow activities are selected from an earlier configured list, and activity results are selected from the list that corresponds to parameters of the configured activity (see "[Workflow Activities](#)").

The result of meeting a rule's conditions is specified in the *Action* field of the form for configuring rules. Possible values:

- "Next Step" – go to the next step of the strategy.
- "Change Step" – go to the strategy step specified in the *New Step* field.
- "Change Strategy" – go to the first step of the strategy specified in the *New Strategy* field.
- "Activity" – execute the activity specified in the *New Activity* field.

The *Priority* field specifies the value determining the order of checking rules (the lower the value, the earlier the rule is checked).

A strategy step can be specified in the *Only for Step* field. If this field is filled in, the rule will be in effect only at the selected step. If the field is left empty, the rule will be applicable for the entire strategy, meaning conditions of the rule will be checked at each step.

When configuration is completed, to confirm the changes, click the [Actions...] button in the strategy schemes form and select "Approve" from the context menu. In the "Date Start" form that opened (see [figure](#)), set the date on which changes become effective (current or future date can be specified), then click the [Proceed] button.

Additional information about the results of approval can be obtained after selection in the user menu item "Full → Process Log → Process Log".

To create a copy of a configured strategy scheme, click the [Actions...] button in the strategy scheme and select "Duplicate" from the context menu.

To delete a scheme from the list, click the [Actions...] button in the strategy schemes form and select "Delete" from the context menu.

The [History] button in the "Strategy Schemes" form (see [figure](#)) opens the "History for <name of strategy>" form together with a group of subordinate forms (see [figure](#)) that contains all previous versions of the selected application workflow strategy and the strategy's version that is currently used by Way4.

The *Date From* and *Date To* fields determine the period for which the strategy version is effective:

- *Date From* – date from which Way4 will use this version of the strategy. This date is specified when changes are applied.
- *Date To* – date from which the next version of the strategy becomes effective. If the field contains "00/00/0000", this means the next version has not been created yet.

History for Application Workflow					<< < > >>		6 of 29		b x	
Date From	Date To	Domain	Name	Code						
01/01/2018	00/00/0000	Application WorkFlow	Application Workflow	APPL_ALL_WF						
01/01/2000	01/01/2000	Application WorkFlow	Application Workflow	APPL_ALL_WF						
01/01/2017	01/01/2017	Application WorkFlow	Application Workflow	APPL_ALL_WF						

Strategies for Application Workflow				<< < > >>		1 of 4		b x	
Name	Code	Is Ready							
Manual Entry Application Strategy	ENTRYSIMPLE	Ready							
File Import Application Strategy	IMPORTSIMPLE	Ready							
Online Load Application Strategy	LOADSIMPLE	Ready							
WS Entry Application Strategy	WSENTRYSIMPLE	Ready							

Steps for Manual Entry Application Strategy				<< < > >>		1 of 4		b x	
Seq Num	Name	Code	Is Ready						
0	Manual entry step	ENTRY	Ready						
1	Accepting Step	ACCEPT	Ready						
2	Declining Step	DECLINE	Ready						
3	Closing Step	CLOSE	Ready						

General Activities for Manual Entry Application		<< < > >>		1 of 2		b x	
Activity	Is Ready						
System Decline Application	Ready						
Check Application	Ready						

Special Activities for Manual entry step		<< < > >>		1 of 4		b x	
Activity	Is Ready						
Approve Application	Ready						
Capture and Edit Application	Ready						
Decline Application	Ready						
Check Application	Ready						

Application workflow strategy history of changes

3.7 Selecting a Strategy

When an application is registered, the system selects a workflow strategy.

These actions are executed on the basis of application parameters. In the Advanced Applications module, sets of parameters are configured with specified values; these sets are called parameter maps.

During configuration, a specific workflow strategy is specified for each parameter map.

If the values of an application's parameters correspond to the values set during map configuration, the strategy specified for the map is selected for the application workflow.

To configure parameter maps, select the menu item "Advanced Applications R2 → Workflow Configuration → Assignment Schemes". The "Assignment Schemes" form together with a group of child forms will open.

Assignment Schemes				<< < > >>	1 of 1	x
Domain	Name	Code	Is Ready			
Application WorkFlow	All Applications	APPL_DISP	Not Ready			

Ins	Del	Query	Approve	Assignments	History
-----	-----	-------	---------	-------------	---------

Assignment for All Applications				<< < > >>	3 of 5	b	x
Priority	Name	Code	Is Ready				
1	Manual Entry	MANUAL	Ready				
2	File Import	IMPORT	Ready				
3	Online Load	ONLINELOAD	Ready				
4	WebService	WS	Ready				
5	Splitted	SPLIT	Ready				

Ins	Del	Query	Parms	Strategy
-----	-----	-------	-------	----------

Parms for Online Load		<< < > >>	1 of 1	b	x
Parameter	Parameter Value				
Creation Type	Online Load				

Ins	Del	Query
-----	-----	-------

Strategy for Online Load		<< < > >>	1 of 1	b	x
Strategy Schema	Strategy				
Application Workflow	Online Load Application Strategy				

Ins	Del	Query
-----	-----	-------

Configuring parameter maps

Parameter maps are grouped in map schemes registered in the "Assignment Schemes" form.

Parameter maps are configured in the "Assignment for <name of map scheme>" form opened from the "Assignment Schemes" form by clicking the [Assignments] button.

A set of parameters with specified values corresponds to each parameter map, as well as a workflow strategy.



If different parameters are specified in one card, they are joined through the logical "AND" operator, different values of the same parameter are joined through the logical "OR" operator.

When selecting a parameter map for an application, a check is made in the ascending order of priorities specified in the *Priority* field.

Parameters of maps are configured and strategies are specified in the child forms opened by clicking the [Parms] and [Strategy] buttons, respectively.

After configuring parameter maps, click the [Approve] button in the "Assignment Schemes" parent form to confirm the changes. In the "Date Start" form that opened (see [figure](#)), set the date on which changes become effective (current or future date can be specified), then click the [Proceed] button.

Additional information about the results of approval can be obtained after selection in the user menu item "Full → Process Log → Process Log".

The [History] button in the "Assignment Schemes" form opens the "History for <name of map>" form together with a group of subordinate forms (see [figure](#)) that contains all previous versions of the parameter map settings and the parameter map's version that is currently used by Way4.

History for All Applications				
Date From	Date To	Domain	Name	Code
01/01/2000	12/02/2018	Application WorkFlow	All Applications	APPL_DISP
12/02/2018	20/06/2018	Application WorkFlow	All Applications	APPL_DISP
20/06/2018	00/00/0000	Application WorkFlow	All Applications	APPL_DISP

Ins Del Query Assignments

Assignment for All Applications			
Priority	Name	Code	Is Ready
1	Manual Entry	MANUAL	Ready
2	File Import	IMPORT	Ready
3	Online Load	ONLINELOAD	Ready
4	WebService	WS	Ready
5	Split	SPLIT	Ready

Ins Del Query Params Strategy

Params for Manual Entry	
Parameter	Parameter Value
Creation Type	Manual Entry

Ins Del Query

Strategy for Manual Entry	
Strategy Schema	Strategy
Application Workflow	Manual Entry Application Strategy

Ins Del Query

History of changes to parameter map settings

The version is selected in the "History for <name of map>" form.

The *Date From* and *Date To* fields determine the period for which the parameter map version is effective:

- *Date From* – date from which Way4 will use this version of the parameter map. This date is specified when changes are applied.
- *Date To* – date from which the next version of the parameter map becomes effective. If the field contains "00/00/0000", this means the next version has not been created yet.

4 Configuring Application Schemes

An application scheme makes it possible to set the main object (client or contract) for which the application is intended and a hierarchy of related objects.

A scheme consists of sections, contains a main section (for entering data about the main object) and set of subordinate sections organised in a hierarchy. Each section is used to enter information about a system object (client, contract, address, etc.). A scheme may consist of one section.

Further when a user creates an application of a selected type according to a scheme, an object structure will be automatically created for the user to enter information about each object.



For a number of objects, information can be added even when the application scheme does not have a section for working with this object (see "[Rules for Building Schemes](#)").

Example of a hierarchical structure of objects:

- Client record (main object).
 - Contract.
 - Address.
 - Card contract.

4.1 Application Section Types

An application scheme is built from sections of various type (see "[Application Schemes](#)").

Section types are registered and configured in the "Appl Section Type" form (Advanced Applications R2 → Setup → Appl Section Type).

Appl Section Type			<< < > >>	1 of 31	X
	Code	Name	Application Object Type		
➔	ADD_INFO	ADD_INFO	Add Info		
	ADDRESS	Address Information	Address		
	AUTH_METHOD	Authentication Method	Auth Method		
	CARD	Card Contract	Contract		
	CHECK_CONTRACT	CHECK_CONTRACT	Contract		
	CLASSIFIER	CLASSIFIER	Classifier		
	CLIENT	Client Information	Client		
	CONTRACT	Account Contract	Contract		
	COUNTERPARTY	COUNTERPARTY	Counterparty		
	CREDIT_LIMIT	Credit Limit	Credit Limit		
	DEVICE	DEVICE	Contract		
	DEVICE_PARMS	DEVICE_PARMS	Device Parms		
	EVENT	EVENT	Event		
	EXTRA_INFO	EXTRA_INFO	Add Info		
	IDENTIFIER	IDENTIFIER	Customer Identifier		
	INSTALMENT	Instalment	Instalment		
	INSTALMENT_LIMIT	Instalment Limit	Instalment Limit		
	LIMIT	LIMIT	Credit Limit		
	MANDATE	Mandate	Counterparty		
	ORDER	Contract Order	Order		
	PARAMETER	PARAMETER	Parameter		
	PREFERRED	Preferred	Preferred		
	RELATED_CARD	Related Card	Contract		
	RELATED_CLIENT	RELATED_CLIENT	Affiliation		
	RELATION	RELATION	Relation		
	STATUS	STATUS	Status		
	STOPLIST	STOPLIST	Stop List		
	SUB_CONTRACT	SUB_CONTRACT	Contract		
	TARIFF	TARIFF	Tariff		
	USAGE	USAGE	Usage		
	WRK_ADDRESS	WRK_ADDRESS	Address		

Section types

The form's fields:

- *Code* – section type code.
- *Name* – the application section type name shown when entering and processing an application.
- *Application Object Type* – the type of object for which this section type is intended.



Special buttons in application processing forms are used to work with a number of application objects. The buttons will only be shown if section types with the corresponding value in the *Code* field are configured. These sections types and their codes are shown in [figure](#). It is not recommended to change or delete these values.



To work with volume-based tariffs, add a record to the "Appl Object Type" form (or make sure the record is present), with the following field values: *Code* = "VOLUMEBASEDTARIFF", *Name* = "VOLUME BASED TARIFF", *Application Object Type* = "Volume Based Tariff".

Use of volume-based tariffs requires an additional agreement with OpenWay.

4.2 Application Schemes

The "Appl Type" form is used to work with application types (Advanced Applications R2 → Setup → Appl Type), see [figure](#).

An application scheme is configured for each application type.

A scheme consists of sections. It contains a main section (for entering data about the main object) and a set of subsections, organized in a hierarchy. The main section of the scheme can be used to work with the "Client" or "Contract" object.

Each section is used to enter information about a system object (client, contract, address, etc.).

Each row in the "Appl Type" form (see [figure](#)) corresponds to an application type. When a type is created, the main object type of the application is specified for it. All other properties of a type are specified when configuring the applications scheme.

Example of a hierarchical structure of objects:

- Client record (main object).
 - Address.
 - Card contract.
 - Contract.



For a number of objects, information can be added even when the application scheme does not have a section for working with this object (see "[Rules for Building Schemes](#)").

Appl Type						
Product Category	Code	Name	Action Type	Object For Type	Is Ready	
Issuing	CREATE_CONTRACT	New_Contract	Add	Client	Ready	
Issuing	NEW_CARDHOLDER	New Cardholder	Add	Client	Ready	
Issuing	NEW_CL_ADDINFO	New Client Add Info	Add	Client	Ready	
Issuing	NEW_LIM	New Limit	Update	Contract	Ready	
Issuing	NEW_USAGE	New Usage	Add	Contract	Ready	
Issuing	UPDATE_CLIENT	Update Client	Update	Client	Ready	
→ Issuing	UPDATE_CONTRACT	Update Contract	Update	Contract	Ready	
Acquiring	NEW_ACQ_CONTRACT	New Acquiring Contract	Update	Client	Ready	
Acquiring	NEW_MERCHANT	New Merchant	Add	Client	Ready	
Acquiring	UPDATE_ACQ_CONTRACT	Update Acquiring Contract	Update	Contract	Ready	
Acquiring	UPDATE_MERCHANT	Update Merchant	Update	Client	Ready	

Sections for Update Contract							
Seq Number	Section Type	Parent Section	Parameter	Action Type Override	Mandatory	Add Data	Is Ready
→ 0	Contract				Yes		Ready
1	AddInfo	Contract			No		Ready
3	Related Card	Contract			No		Ready
4	Status	Contract			No		Ready
5	Stop List	Contract			No		Ready

Application schemes

The "Appl Type" form's fields:

- *Product Category* – category to which the application type belongs ("Issuing" or "Acquiring").
- *Name* – application type name.
- *Code* – application type code.
- *Action Type* – type of action with the application object.
- *Object For Type* – type of main object for which the application is used: "Client" or "Contract". The field may be empty if an application scheme is set up to create a hierarchy of contracts with different clients (see ["Examples of Creating Application Schemes with a Hierarchy of Contracts for Different Clients"](#)).
- *Is Ready* – indicates whether the application type is ready for use. The value of the field will be "Yes" after the application type has been approved.

The {Validate} command is used to validate the scheme belonging to the selected application type. If errors are found during validation, the corresponding message will be displayed.



It is not recommended to delete the "New Cardholder" and "New Merchant" types from the list, or to change their codes.

Work with an application scheme takes place in the "Sections for <scheme name>" child form opened by clicking the [Sections] button in the parent form (see [figure](#)).

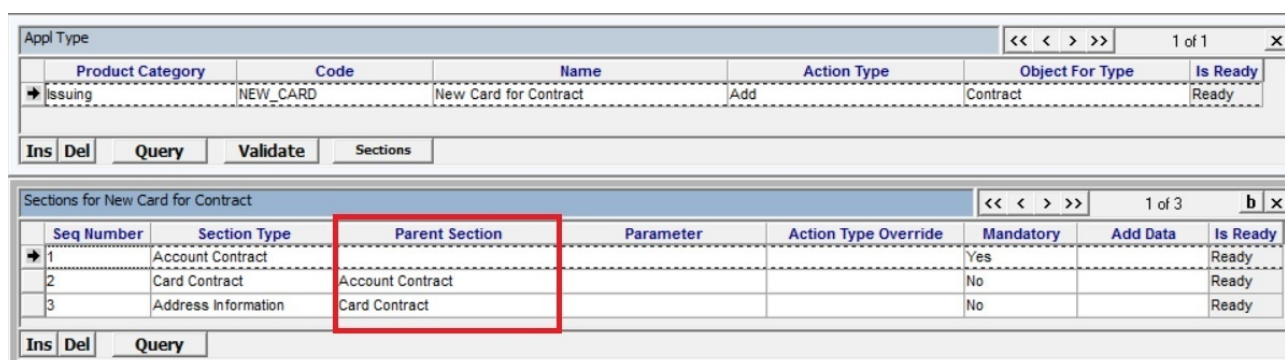
The form's fields:

- *Seq Number* – ordinal number of a section in the scheme. The field must contain a unique value in the scheme.

- *Section Type* – section type (see "Application Section Types"). The main section of the scheme must be used to work with either a "Client" or "Contract" object.
- *Parent Section* – type of the parent section. This field can be used to create hierarchical structures of sections.
- *Parameter* – parameter determining the object type (see the table). The parameter code is specified in this field.
- *Action Type Override* – action type that is used to redefine the action type that is set at the application type level (*Action Type* field of the "Appl Type" form).
- *Mandatory* – indicates whether it is mandatory to fill in this section in the application.
- *Add Data* – additional information. For sections with the "Contract" types, this field may specify the MULTIPLE tag making it possible to use sections to add several contracts at one level of a hierarchy to an application.

Example of a hierarchical structure of objects (see figure):

- Account contract (main object).
 - Address.
 - Card contract.



Product Category	Code	Name	Action Type	Object For Type	Is Ready
Issuing	NEW_CARD	New Card for Contract	Add	Contract	Ready

Seq Number	Section Type	Parent Section	Parameter	Action Type Override	Mandatory	Add Data	Is Ready
1	Account Contract				Yes		Ready
2	Card Contract	Account Contract			No		Ready
3	Address Information	Card Contract			No		Ready

Object hierarchy

The table below shows possible parameter values for different types of object.

Possible parameter values depending on object type

Object type	Possible parameter values
Address	Address type code (Full → Configuration Setup → Client Classifiers → Address Types).
Auth Method	<p>Authentication type code (Authentication Module → Authentication Types).</p> <p>For this parameter, the value must be specified at the section type configuration level.</p>

Object type	Possible parameter values
Standing Order	<p>The standing payment order code.</p> <p>For this parameter, the value must be specified at the section type configuration level. The value of the payment type code can be specified (PAYMENT_TYPE=<payment type code>).</p>
Usage Limiter	Usage limiter code (Full → Configuration Setup → Products → Service Packs → [Usage]).
Tariff	<p>The tariff type and tariff code can be specified: TARIFF_TYPE=<tariff type>;TARIFF_CODE_EXT=<tariff code>; (Tariffs → Tariff Types & Tariff Domains → Tariff Domains).</p> <p>The tariff role is defined according to the tariff type.</p> <p>If no value is specified for the "Tariff" parameter, the parameter is interpreted as a tariff role.</p>
Credit Limit	<p>Credit limit type:</p> <ul style="list-style-type: none"> • "F" ("Financial") – credit limit. • "N" ("NonFinancial") – additional authorization limit.
Instalment	Instalment scheme code (Instalments → Instalment Configuration → Instalment Schemes, <i>Service Code</i> field value).
Instalment Limit	Contract instalment limit type code (Instalments → Instalment Configuration → Instalment Scheme Groups → Instalment Limit Types).
Device parameters	Code of the channel for which device parameters will be redefined (Full → Configuration Setup → Main Tables → Message Channels).
Classifier	Client or contract classifier code (Full → Configuration Setup → Common Handbooks → User Classifiers).
Parameter	Client or contract custom parameter code (Full → Configuration Setup → Common Handbooks → Contract Parameters Setup).

Object type	Possible parameter values
Contract	Contract category: <ul style="list-style-type: none"> • "M" ("Merchant") – device contract. • "A" ("Account") – account contract. • "C" ("Card") – card contract.
Counterparty	Payment type code (Full → Configuration Setup → Transaction Types → Payment on Account Types).
Event	Event type code (Full → Configuration Setup → Products → Issuing Private Products (Issuing Corporate Products) → Issuing Event Types; Full → Configuration Setup → Products → Acquiring Products → Acquiring Event Types).

4.3 Rules for Building Schemes

When building an application scheme, the following rules must be observed:

- The type of section for working with a "Client" or "Contract" object can be used one time in the scheme. If it is necessary, for example, to include two sections with the "Contract" type in a scheme (for a merchant contract and a related device contract), two section types for working with the "Contract" object must be created in advance.
- Section types for working with all other object types must be unique within their "branch" of a scheme.
- When working with an application, it is possible to enter information about a number of objects, even if a special section has not been created for them. Forms for working with these objects are available from the form of the parent object – client ("Client Info") or contract ("Contract Info").

The following is a list of such objects:

- Classifier.
- Credit Limit – for contract only.
- Instalment Limit – for contract only.
- Event – for contract only.
- Usage Limiter – for contract only.
- Tariff – for contract only.
- Standing Order – for contract only.
- Address.
- Affiliation.
- Parameter.

- Contract (if the parent object is an issuing account contract).
- Card (if the parent object is a card contract).
- Device Rec (if the parent object is an acquiring module contract).
- Sections for working with a "Client" or "Contract" object type can act as the parent to sections of another type. Possible combinations of section object types are shown in [table](#).
- Two or more sections that are intended to work with this object type and have the same parameter value cannot be used for the following object types in one parent section:
 - Address.
 - AddInfo.
 - AuthMethod.
 - Classifier.
 - Device Parameters.
 - Event.
 - Parameter.
- It is possible to build a scheme consisting of one section. In this case, the main section can be used to work with any type of object. Here it should be noted that child objects cannot later be added to such a scheme (if the main section is used to work with an object type other than "Client" or "Contract").

Possible combinations of section object types

Parent section's object type	Possible child section object types	Issuing	Acquiring
Client	AddInfo	+	+
	Address	+	+
	Affiliation	+	
	AuthMethod	+	
	Classifier	+	+
	Contract	+	+
	Customer Identifier	+	
	Counterparty	+	

Parent section's object type	Possible child section object types	Issuing	Acquiring
Contract	AddInfo	+	+
	Address	+	+
	Affiliation	+	+
	AuthMethod	+	
	Classifier	+	+
	Preferred	+	
	Client	+	
	Card	+	+
	Contract	+	+
	Credit Limit	+	
	Instalment	+	
	Instalment Limit	+	
	Customer Identifier	+	
	Device parameters		+
	Device Rec		+
	Event	+	+
	Standing Order	+	+
	Counterparty	+	+
	Relation	+	+

Parent section's object type	Possible child section object types	Issuing	Acquiring
	Status	+	+
	Stop List	+	
	Tariff	+	+
	Usage Limiter	+	+

4.4 Examples of Creating Application Schemes with a Hierarchy of Contracts for Different Clients

Way4 allows application schemes to be constructed with a multilevel hierarchy of contracts for different clients.

Application schemes for each client (section with the "Client" object type) that is added as a child section must:

- Contain a child section with the "Contract" object type and "Add" action type. An action type can be set in the application type (in the *Action Type* field of the "Appl Type" form) or redefined in a section in the application scheme (*Action Type Override* field of the "Sections for <scheme name>" form).
- Allow an indicator to be set specifying that the section is optional. If the "Client" section is marked as optional ("No" in the *Mandatory* field) and no client identifiers (*Registration #*, *Client Number*, *Social Security #* or *Individual Number*) are specified in the application, this section of the application and all its child sections will be ignored.

Examples of scheme construction are shown below.

Example 1: Creating a hierarchy of contracts with different clients.

Create a hierarchy of contracts for two different clients: an account contract (client 1) and a subordinate card contract (client 2).

The object hierarchy can be shown as follows:

- Client 1 (main section in the scheme)
 - Account contract
 - Client 2
 - Card contract for client 2

Application scheme setup according to this example:

Appl Type					<< < > >>	1 of 1	x
Product Category	Code	Name	Action Type	Object For Type		Is Ready	
Issuing	ADD_TWO_CLIENTS	Add Two Clients	Add			Ready	
Ins	Del	Query	Validate	Sections			
Sections for Add Two Clients					<< < > >>	4 of 4	b x
Seq Number	Section Type	Parent Section	Parameter	Action Type Override	Mandatory	Add Data	Is Ready
1	Client Information				Yes		Ready
2	Account Contract	Client Information			No		Ready
3	Client 2 Information	Account Contract			No		Ready
4	Card Contract	Client 2 Information			No		Ready
Ins	Del	Query					

Creating a hierarchy of contracts with different clients

The *Object For Type* field to specify the application's main object type must be empty in cases for two different clients, like the one in this example.

To include two sections with the "Client" type in the application scheme, two separate types of section must have been created in advance. In [figure](#), these are the "Client Information" and "Client 2 Information" sections. For information about creating section types, see the section "[Application Section Types](#)".

"Client Information" is the main section in the scheme ("Client" object type). An account contract ("Contract" object type) is added as a child section for this client. The second client is linked to the first client's account contract. The second client's card contract is a subcontract of the account contract.

Example 2: Creating a subcontract for a new client who differs from the parent contract's client.

The client (client 1) already has an account contract in Way4. A new card contract must be created for another client (client 2).

An object hierarchy is created for the account contract in Way4 and can be shown as follows:

- Client 2 (main section)
 - Card contract for client 2

Application scheme setup according to this example:

Appl Type					<< < > >>	1 of 1	X
Product Category	Code	Name	Action Type	Object For Type	Is Ready		
Issuing	ADD_SUBCLIENT	Add Sub-Client with Card	Add	Contract	Ready		
Ins Del Query Validate Sections							

Sections for Add Sub-Client with Card					<< < > >>	1 of 2	b X
Seq Number	Section Type	Parent Section	Parameter	Action Type Override	Mandatory	Add Data	Is Ready
1	Client Information				Yes		Ready
2	Card Contract	Client Information			No		Ready
Ins Del Query							

Creating a subcontract for a new client who differs from the parent contract's client

The application's main object type is specified in the *Object For Type* field – "Contract". The account contract is the main contract for all contracts created for the new client.

"Client Information" is the main section in the scheme ("Client" object type). A card contract ("Contract" object type) is added as a child section for this client.

All contracts for client 2 are subordinate to the specified account contract.