

#### CONTENTS



# 

J J 💱 J J 🔁 Ý J J 🍨 🔹 . 🔷 ها ها با با بن بن با بن به ها ها

0

0















## **RISK MONITORING SERVICE**

The central feature of the NYCE® Risk Management Solutions is the Risk Monitoring Service (RMS), a service created in association with FICO®. RMS is a real-time solution that analyzes incoming transactions to help detect suspicious debit card activity. Using information in the specific transaction, terminal and cardholder historical profiles, RMS creates three unique Falcon Fraud scores for every transaction: card, terminal and a combined score.

#### NYCE's Risk Management Solutions Standard Package includes:

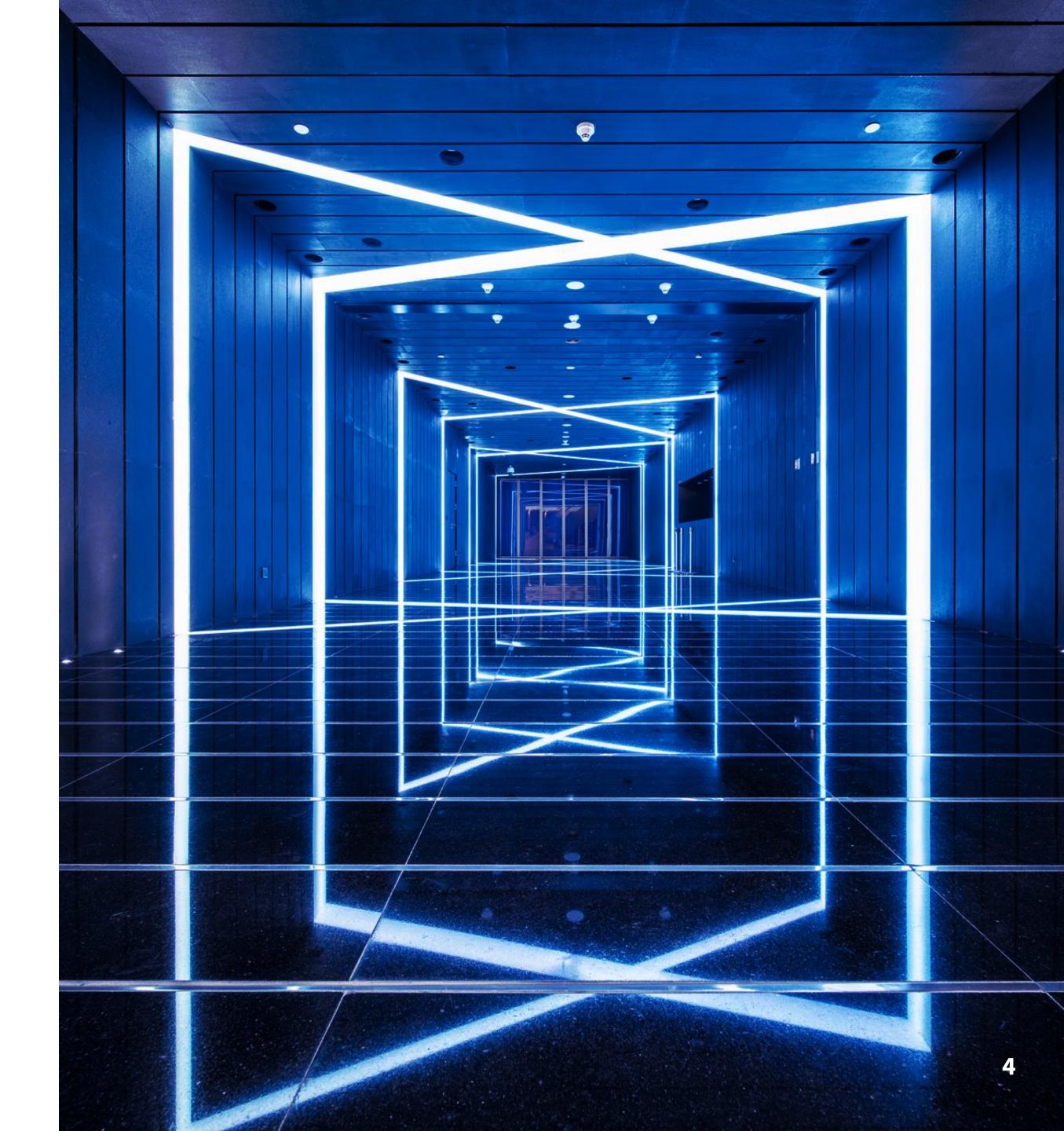
- Risk Monitoring Service
- Card Alert Service
- iCVx Verification Service
- Authorization Blocking
- SOC1-Type2 Report



#### HOW DOES FALCON FRAUD SCORES WORK?

Scores range from 000 to 999; the higher the score, the greater the likelihood of fraud. With a flexible rules engine, RMS uses these scores and other transaction attributes to make real-time decisions to potentially deny fraudulent transactions as they occur.







### **CARD ALERT SERVICE**

Another core feature of NYCE's Risk Management Solutions standard package is the FICO Card Alert Service, a sophisticated transactionmonitoring service that leverages consortium data to provide "point of compromise" identification when fraud occurs. Card Alert receives fraud reporting from multiple EFT networks, including NYCE and reports suspected counterfeit cards to institutions prior to losses occurring. The Card Alert Service is also fully integrated with the NYCE RMS platform, making this industry standard an even more robust tool.



### AUTHORIZATION BLOCKING SERVICE

NYCE's Authorization Blocking Service gives issuers the ability to block transactions where fraud "events" occur in a specific pattern (e.g., purchases being performed in specific country codes). A block denies all transactions matching the specific criteria. Since this type of blocking affects both legitimate and fraudulent transactions, a block generally remains in place only until the fraud event has run its course. NYCE clients may request one complementary Authorization Blocking event per calendar year.



				• the contribute colors its cal = contr') mirror_mod = modifier_ob.modifier	inderor sirr
	KOR")			antine antes 110 million aufent	s.new("million
		dir termen melliftigt innen andres	annelle ber ebinete ite best mit	• the and the ob.modifier	
	# set mirror #	A STATE OF A	<pre>cob.modifiers.new("mirror_mirror","MIRROR") </pre>	mirror_mod = moultine	
	Mirron Mi	to mirror ob OPERATOR CLASS		<pre># set mirror object to mirror_ob # set mirror mod.mirror_object = mirror_</pre>	ob
	Mirror mod	or object mirror mod = modifier	oh medifiere med"-immer minner" "MIRROR")	# set mirror object to mirror.	
	<pre>mirror_mod = modifier_objectiondifier operationdifier # set mirror_object to mirror_mod.use mirror_mod.mirror_object = mirror_modecides</pre>	ob	_ob.modifiers.new( mirror_mirror, rither )	<pre># set mirror object to mirror_ob mirror_mod.mirror_object = mirror_</pre>	
	<pre># set mirror_object to mirror_mod.use_ mirror_mod.mirror_object_mod.use_ mirror_mod.mirror_object_mod.use_ if _operation == ^MIRROR_operation == ^MIRROR_operation == ^MIRROR_operation == ^MIRROR_operation_mirror_mod.use_y mirror_mod.use_y mirror_mod.use_ mirror_mod.use_y mirror_mod.use_ mirror_mod.use_y mirror_mod.use_ mirror_mod.use_y mirror_mod.use_ mirror_mod.use_y mirror_mod.use_ mirror_mod.use_x mirror_mod.use_ elif_operation == ^MIRROR_vation == mirror_mod.use_x mirror_mod.use_y</pre>	# set mirror object to mi		TTEROR X":	
	mirror modeuse	X = True	<pre>ject = mirror ob mirror med - medifier of medifiers.med("mirror mirror", "MIRROR")</pre>	if _operation == "MIRROR_X":	
	mod.mirror object mod class	MinrocX( boy if . operation == ."MTRROP	Vn = set mirror object to mirror ob	Timon' MIRRORC)d. use X = That	
	if operation alic modilizer	This adds an y in perator):	<pre>pect = mirror_ob     mirror_woil=modifier_ob_modifiers.mew("mirror_mirror","MIRROR") X", ****t mirror_object to mirror_ob     mirror_object = mirror_ob     mirror_mod = modifier_ob_modifiers.mew("m     """     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     ""     """     """     """     """     """     """     """     """     """     """     """     """     """"     """"""</pre>	<pre>mirror_mod.use_x = False mirror_mod.use_y = False</pre>	
	mirror "MIRROR OPERation == "	Midnamer "objects and & mirror to the	selected object""	<pre>mirror_mod.use_y = False mirror_mod.use_z = False elif_operation == "MIRROR_Y": nodifiemsroeumod.use_x) = False or ", "MI mirror_mod.use_y = True mirror_mod.use_z = False mirror_mod.use_z = False 7";</pre>	
	mirror_mod.use_x = True_mod.use	label - "M: UL operation - "INTER	OF X" is set and mirror mod.mirror_object = mirror_ob	elif operation == Plinter Palseor, Pl	
	mirror mod.use y = Falso mod.use	$V = True mirror_mod_use x = Fa$	In T information and use y Filst alse mirror and use y True alse mirror and use y True	nodifiemspride/whod/userx True	
	elif opportion use z talso mod.use	bclassmethod	rue area mirror mod.use x = True	mirror_mod.use_y False	
	mirpor	Terror mod.use_z = Fi	alse arroy adducts Par mod.use y = False	mirror_mod.use_y mirror_mod.use_z = False 	
	<pre>mirror_mod.use_ymirror_mod.use_ mirror_mod.use_zmirror_mod.use_ elifoperation</pre>	x - 10 thun cont "Irror mod.use x = F	OR Z": mirror_mod.use_z = False alse_mirror_elif_operation == "MIRROR_Y":	ob elif_operation == "MIRROR_Z":	
			t is not None or mod use x = Falser object	<pre>elif_operation mirromirror_mod.use_x = False mirror_mod.use_y = False mirror_mod.use_y = True</pre>	
	elif on a way a wa	7 - True	million mouruse y = 11 de	mirror_mod.use_j mirror_mod.use_z = True	
gets the modific	elif _operation == "MIRROR_Z": mirror_mod.use_x == "MIRROR_Z": ss its not a minror_mod.use_irror_adseselect= mirror_mod.use_irror_adseselect=	#selection at the er	<pre>mirror_mod.use_z = False elif_operation == "MIRROR_7":</pre>	s not a well") mirror_mou.usc_2	
inc mouttier unle	ss its not a minner mod.use x #Falsetion at	mirror_ob.select= 1	mirror mod.use_x = False		
	<pre>mirror_mod.use_x =# Palsetion at ss its not a minror_mod.useirror_adseselect= mirror_mod.useirror_adseselect=</pre>	1 bpv.context.scene.object	mirror modifiermoduse of False = True	mirror_ob.select= 1	clas
	fier object	=1 print("Selected" + str(		mirror_ou.serece	Lifion ob
			modifier on TCCCF MODIUSE y False #selection at the end -add back the des mirror ob.select= 1	<pre>elected mixing imprivation object=1</pre>	
	#selection at phinte("Selected"ka mirror_ob.select= 1 #mirror_ob.select=	tstn(modifiler ob)) at modifier of	<pre>modifier_ob.select=1 "MTRROR_Y"</pre>		
	mirror_ob.select= 1 #mirror_ob.sel	lect = 0 except:	<pre>oject the bpy.contextiscene.objects.active = modifier</pre>	fier ob is the active ob the select = 0	
	E DOV CODIEN		t exactly wo himror ob.select = 0		
	<pre>ve b bpy.context.scene.mbjects.activects print("Selectextceptstr(modifier_ob) #mirror ob.select.scene.top)</pre>	odifienco select = 1 OPERATOR CLAS	<pre>#one = bpy.context.selected_objects[0] @</pre>		
	#mirror ob colect	) # modifier ob is the active ob	<pre>#bpy.data.objects[one.name],select = 1 se except:</pre>	except:	h")the the last
	<pre>#one = bny context colors</pre>	select exactly two objects, the	a last one gets the modifier unless its	except: ts. the last one gets the modifier unless its not a mes not a mesh )	" Jojecus, the las
	<pre>#bpy.data.objects[one.name]opedator except: # Minnon Teal</pre>		# Mirror Jool mirror mod.use y = False		
the last one ge	ts the modification logi	mis adds an X mirror to t	ne selected object mor mod use y - reise		
	except: # Mirror Tool ts the modif <b>print("please select</b> exactly"to	the last one gets the	modifier unless its not a mesh") True		
	CONTRACTOR CLASSES				



#### ICVX VERIFICATION SERVICE

To enhance your risk mitigation strategy, you may opt to have NYCE perform iCVx (EMV contact chip), CVx (magstripe) and CVx2 (eCommerce) verification on your behalf or in addition to your validation and authorization. Counterfeit cards created using phished values will generally fail NYCE's iCVx verification process.

@classmeth def poll(«

Mirror)



### SOC1-TYPE 2 (SSAE18) REPORT

The System and Organization Controls (SOC) for Service Organizations Report is prepared by independent auditors and provides a description and opinion of the control structure of FIS as it relates to the NYCE Payments Network. Access the most recent report on the FIS Client Portal.







#### NYCE RISK MANAGEMENT SOLUTIONS STANDARD PACKAGE

Service	Core Offering	
Risk Monitoring Service	X	
Card Alert Service	X	
iCVx Verification Service	1 free event annually	
Authorization Blocking	X	
SOC1-Type2 Report	X	

**Note:** Authorization Blocking and iCVx Verification Service are enabled at the client's direction. Specific rules and rule parameters are created and customized at the client's discretion. Contact your NYCE representative for additional information and details on service activation, rule creation and pricing.



# CONTACT US

For more information about NYCE Risk Management Solutions, contact your NYCE representative:

800.822.6758

......................

......

......

......

........

.........

......

......

...........

......

...............

......

.......

----

......

......

...

...

....

....

..... .....

......

......

.......

·····

\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*

...........

..........

........

......

......

.....

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

www.nyce.net 

...........

.....

.....

.....

.....

...........

..........

.....

.........

........

.......

#### **About FIS**

2000

1000

....

000

000

...

.... 

.....

.....

.....

.....

.....

.....

.....

.....

FIS is a leading provider of technology solutions for merchants, banks and capital markets firms globally. Our more than 55,000 people are dedicated to advancing the way the world pays, banks and invests by applying our scale, deep expertise and data-driven insights. We help our clients use technology in innovative ways to solve business-critical challenges and deliver superior experiences for their customers. Headquartered in Jacksonville, Florida, FIS is a Fortune 500<sup>®</sup> company and is a member of Standard & Poor's 500<sup>®</sup> Index.







ΪM linkedin.com/company/fis





©2023 FIS FIS and the FIS logo are trademarks or registered trademarks of FIS or its subsidiaries in the U.S. and/or other countries. Other parties' marks are the property of their respective owners. 2292694