Release Notes for SAS® Fraud Management 4.4_M1, Hot Fix 7
Version 5, Release 13

Description	Component	Summary and Business Impact	Test Scenario
On the mainframe, sensitive data is not hidden or masked in the Universal SAS Connector (USC) audit log file.	USC	**Summary:** When the USC logging level is high, or when error messages are printed at any log level, sensitive data is written to the audit log file. **Business Impact:** Sensitive data that is written to audit logs in clear text can be compromised.	After you apply the hot fix, you can remove sensitive information from the USC audit log messages by editing the USCFDICT file. You can remove the XML tag(s) that are placeholders for the data that you want to remove. For example, the tag <TXT6> usually contains the card, customer, or account number, depending on the message in the USCFDICT file. You can remove this tag to prevent the value from being written to the audit log.
On Oracle systems, performance might be slow for the queries that are used in rule estimation.	ESTIMATION	**Summary:** On Oracle systems, queries that are used by rule estimation might be slow. **Business Impact:** Rule writers are less productive because it takes longer for them to develop new rules or to modify existing rules so that rules can identify more fraudulent transactions.	After you apply the hot fix, the DECODE_BASE64_SIGNATURE function in the Transaction Data Repository (TDR) database includes performance improvements. As a result, estimation performance might improve.
Segment-variable data in the User Authorization Request (RUA) is not displayed in the **Related Rows** grid.	RULES and EXPLORE	**Summary:** When RUA variables are added to the **Related Rows** columns on the **Explore** tab and in the rule editor, the values are not displayed. *(continued on next page)*	After you apply the hot fix, the RUA variable values are displayed in the **Related Rows** grid on the **Explore** tab and in the rule editor.

Description	Component	Summary and Business Impact	Test Scenario
		Business Impact: Analysts have difficulty servicing alerts because they cannot validate the transaction data without reviewing the RUA values.	
On the mainframe, an OC4 abend (ABEND0C4) occurs during an Online Scoring Engine (OSE) restart.	OSE (Mainframe)	**Summary:** The OSE receives an 0C4 abend during OSE initialization. The abend is caused by an incorrect calculation for the controller lookup-list table. **Business Impact:** The OSE fails to start and transactions are not scored. Fraudulent transactions are not identified while the engine is not running.	After you apply the hot fix, the OSE starts successfully.
When you create a new user by copying an existing user, the multi-organization selections are not copied.	ADMIN	**Summary:** On the **Users** tab, you can create a new user by selecting an existing user and clicking the **Copy** button. The new user is created successfully, but no multi-organizations are assigned. **Business Impact:** Administrators might require more time to create new users if the multi-organizational structure is complex. The administrator must manually update every new user to select the multi-organizations.	After you apply the hot fix, the multi-organization selections are correct for users that are copied from existing users.
The SMH_MULTI_ORG_NODE_KEY value is not displayed in the transaction grid on the **Explore** tab.	EXPLORE	**Summary:** The SMH_MULTI_ORG_NODE_KEY value for the transactions that are listed in the **Related Rows** grid on the **Explore** tab are blank. **Business Impact:** Users review transaction data using the **Explore** tab. When data is not displayed, it might impact users' ability to service alerts and test rules.	After you apply the hot fix, the SMH_MULTI_ORG_NODE_KEY value is displayed in the transaction list on the **Explore** tab.

Description	Component	Summary and Business Impact	Test Scenario
On DB2 systems, time-field values for transactions in the **Related Rows** grid do not match the Transaction Data Repository (TDR) database.	RULES and EXPLORE	**Summary:** Time values in DB2 are not stored in Coordinated Universal Time (UTC) like other date-time values. The time values are incorrectly shifted by a number of hours, based on your time zone, before they are displayed in the **Related Rows** grid on both the **Rules** and **Explore** tabs. The values do not match the values that are stored in the TDR database. Two examples of time fields that are incorrect in the grid are the following: • RQO_TRAN_TIME • RQO_TRAN_TIME_ALT **Business Impact:** Users see incorrect values for some time fields in transactions that are displayed on the **Rules** and **Explore** tabs. When data is not correct, it might impact users' ability to service alerts and write effective rules.	After you apply the hot fix, the time values are displayed correctly in the **Related Rows** grid.
The Alert Destination selection list is limited to 100 queues.	RULES and EXPLORE	**Summary:** On the **Explore** tab, you can create an alert for a transaction in the grid by right-clicking the transaction and selecting **Create an Alert** from the menu. In the Create Alert dialog box, the **Alert Destination** drop-down contains a list of active queues. That list contains the first 100 queues in alphabetical order. If there are more than 100, you cannot create an alert on those additional queues. When you edit a rule using the Guided Approach, the **Alert Destination** drop-down is also displayed when you click the **Create an Alert** check box. You encounter this same issue when there are more than 100 available queues. You can only select from the first 100 queues. *(continued on next page)*	After you apply the hot fix, the Alert Destination drop-down will display up to 100,000 queues.

Description	Component	Summary and Business Impact	Test Scenario
		Business Impact: If more than 100 queues are available, then both rule writers and analysts are unable to select some of the queues on the **Explore** tab or when editing rules created using the Guided Approach.	
There is a reflective cross-site scripting (XSS) vulnerability in the alert search function.	ANYLSTSWORK	**Summary:** When you search for alerts using the **Demographic Search** function on the **Alerts** tab, the **Name** and **Address** fields are vulnerable to cross-site scripting when you select the **Starts With** option. **Business Impact:** An attacker can inject malicious code, which then can execute in your browser session. It is possible for malicious code to be injected into the JavaScript code for two fields in the **Demographic Search** dialog. Then, someone can execute added code in the authenticated user's browser session, placing sensitive data at risk of being compromised.	After you apply the hot fix, the **Name** and **Address** fields in the **Demographic Search** function do not run any JavaScript. As a result, the fields are no longer vulnerable to a cross-site scripting attack.
On Oracle systems, the millisecond values for the RQO_TRAN_TIME and RQO_TRAN_TIME_ALT fields are not preserved when a transaction is inserted into the Transaction Data Repository (TDR) database.	ODE	**Summary:** The SAS OnDemand Decision Engine does not preserve the millisecond values for RQO_TRAN_TIME and RQO_TRAN_TIME_ALT fields when the transaction is inserted into the Transaction Data Repository (TDR) database. Instead, the millisecond value is always **`000`**. Earlier releases of SAS Fraud Management included the millisecond value in the time fields in Oracle. **Business Impact:** During peak periods, many transactions can be received in a second. The millisecond values for these fields are required by some customers to ensure correct ordering of transactions when they review alerts. For example, without the millisecond value, the denial of the transaction might appear to have arrived before the pre-authorization.	After you apply the hot fix, the OnDemand Decision Engine preserves the millisecond value for the RQO_TRAN_TIME and RQO_TRAN_TIME_ALT fields in the transactions when the transactions are inserted into the TDR.

Description	Component	Summary and Business Impact	Test Scenario

The REST API methods that update an alert status (by alert ID or entity value) fail.	REST API	**Summary:** If you specify a value for the **`systemBlockCode`** field when you give an assessment to an alert, the REST API call fails. When the failure occurs, the following response is returned:
		
		`{`
		`"errorCode": 3,`
		`"message": "Unexpected failure occurred.",`
		`"details": [],`
		`"remediation": "",`
		`"links": [],`
		`"version": 1,`
		`"httpStatusCode": 500`
		`}`
		
		These two methods are impacted:
		• **`/alerts/specifiedEntity/actions/status`**
		• **`/alerts/{alertId}/actions/status`**
		
		Business Impact: If you use the REST API methods to update alert status, the update fails. The failure occurs when a system or transaction block code is specified. If a block is not applied, subsequent transactions, possibly fraudulent, are not be blocked for the entity.
			After you apply the hot fix, the update of alert status by using the REST API completes successfully.